Multistage Stochastic Optimization: From Optimal Transport-Based Scenario Tree Reduction to Robust Optimization

A dissertation submitted by

Daniel Mimouni

in satisfaction of the requirements for the
PhD Thesis

École Doctorale 580
Sciences et Technologies de l'Information et de la Communication (STIC)

on
October 24, 2025

Jury members:

Franck Iutzeler	Professor, Institut Mathématique de Toulouse	Reviewer
Alois Pichler	Professor, University of Technology, Chemnitz	Reviewer
Delphine Bresch-Pietri	Associate Professor, Mines Paris – PSL	Jury
Paul Malisani	Research Engineer, IFPEN	Supervisor
Jiamin Zhu	Research Engineer, IFPEN	Co-Supervisor
Welington de Oliveira	Associate Professor, Mines Paris – PSL	Thesis Director
Michel De Lara	Professor, École Nationale des Ponts et Chaussées	Jury

À mes parents,
dont le plus beau cadeau a été
de me faire grandir avec Elsa et Nathan.

Preface

This document compiles the work accomplished during the three years of my PhD, it has been the result of close collaboration with Paul Malisani, Jiamin Zhu, and Welington de Oliveira. This manuscript is structured into four primary sections: one detailing our research on a novel method for computing *Wasserstein Barycenters* (WB), a second on extensions of this methods to more complex cases of *constraint barycenters*, a third presenting our proposed approach to addressing *scenario tree reduction via WBs* and finally a complete pipeline to optimize an *industrial Energy Management System*. Each of these parts represents an independent scientific contribution. Considered together, they complement and articulate coherently to provide an effective response to the applied research problem of energy management.

Introduction for non-specialists

This subsection illustrates, through a simple example, how scenario trees and decision-making under uncertainty are connected.

Intermittent power sources, such as solar farms, must follow grid integration strategies. Consider a solar panel farm supplying electricity to a city. The goal is to meet the city's energy demand at all times. However, solar power generation depends on unpredictable factors like sunlight and weather. To prevent shortages during low sunlight periods, the system must be supplemented by more stable electricity sources—for example, nuclear power, which is common in France.

According to European market regulations, electricity providers must forecast and plan their production and purchases one day in advance. The solar farm manager must therefore estimate the expected production of the farm and the city's electricity consumption for the next day, both represented as probability distributions. Based on these forecasts, the manager decides how much electricity to purchase from external sources.

Then, throughout the day and at each time step (e.g., every 30 minutes), the manager observes the actual production and consumption. Decisions must be updated dynamically to maintain the balance between supply and demand. Three situations can arise:

- **Shortfall**: If consumption exceeds the sum of solar production and the initially purchased external electricity, the manager must buy additional electricity at a higher spot-market price.
- **Balance with surplus**: If the combined production and external supply meet or exceed the demand, the excess electricity can be stored (e.g., in dams or batteries) for future use.
- **Strategic purchase**: Even when supply matches demand, the manager may choose to purchase and store additional electricity if forecasts suggest that doing so now is cheaper than buying it later.

This example highlights how each decision affects future possibilities, forming a chain of interconnected decisions. The overarching goal of the manager is to minimize total costs over time. This thesis addresses the following question:

What strategy should the manager adopt, given that the future is uncertain and that past observations may differ significantly from future events?

Préface

Ce document rassemble les travaux réalisés au cours des trois années de ma thèse, il est le fruit d'une collaboration étroite avec Paul Malisani, Jiamin Zhu et Welington de Oliveira. Ce manuscrit est structuré en quatre parties principales : la première présente nos recherches sur une nouvelle méthode de calcul de *barycentres de Wasserstein* (BW), la deuxième porte sur les extensions de cette méthode à des cas plus complexes de *barycentres contraints*, la troisième expose notre approche proposée pour la *réduction d'arbres de scénarios via BWs*, et enfin, la dernière décrit une chaîne complète de traitement permettant d'optimiser un *système de gestion de l'énergie* industriel. Chacune de ces parties constitue une avancée scientifique indépendante. Considérées ensemble, elles se complètent et s'articulent de manière cohérente afin d'apporter une réponse efficace au problème énergétique étudié.

Introduction pour les non-spécialistes

Cette sous-section illustre, à travers un exemple simple, comment les arbres de scénarios et la prise de décision en situation d'incertitude sont liés.

Les sources d'énergie intermittentes, comme les fermes solaires, doivent s'intégrer au réseau électrique selon des stratégies spécifiques. Considérons une ferme de panneaux solaires qui alimente une ville en électricité. L'objectif est de satisfaire à tout moment la demande énergétique de la ville. Toutefois, la production solaire dépend de facteurs imprévisibles comme l'ensoleillement et la météo. Pour éviter les pénuries lors de faibles périodes d'ensoleillement, le système doit être complété par des sources d'électricité plus stables — par exemple, l'énergie nucléaire, très présente en France.

Selon les règles du marché européen, les fournisseurs d'électricité doivent prévoir et planifier leur production et leurs achats la veille pour le lendemain. Le gestionnaire de la ferme solaire doit donc estimer la production attendue de la ferme ainsi que la consommation électrique de la ville, toutes deux représentées par des distributions de probabilité. À partir de ces prévisions, il décide de la quantité d'électricité à acheter auprès de sources externes.

Ensuite, au fil de la journée et à chaque instant (par exemple toutes les 30 minutes), le gestionnaire observe la production et la consommation réelles. Il doit alors adapter ses décisions dynamiquement pour maintenir l'équilibre entre l'offre et la demande. Trois situations peuvent se présenter :

- **Déficit** : si la consommation dépasse la somme de la production solaire et de l'électricité externe initialement achetée, le gestionnaire doit acheter de l'électricité supplémentaire à un prix plus élevé sur le marché spot.
- **Équilibre avec surplus** : si la production combinée et l'approvisionnement externe couvrent ou dépassent la demande, l'excédent peut être stocké (par exemple dans des barrages ou des batteries) pour une utilisation future.
- **Achat stratégique** : même lorsque l'offre correspond à la demande, le gestionnaire peut choisir d'acheter et de stocker de l'électricité supplémentaire si les prévisions indiquent qu'il sera moins coûteux de le faire maintenant que plus tard.

Cet exemple met en évidence le fait que chaque décision influence les décisions futures, formant ainsi une chaîne de choix interconnectés. L'objectif global du gestionnaire est de minimiser les coûts totaux sur l'ensemble de la période. Cette thèse répond à la question suivante :

Quelle stratégie le gestionnaire doit-il adopter, sachant que le futur est incertain et que les observations passées peuvent être très différentes des événements à venir ?

Remerciements

Je souhaite tout d'abord remercier les personnes qui ont accepté d'évaluer le travail réalisé au cours de cette thèse. Je remercie tout particulièrement les rapporteurs, le Pr. Alois Pichler, dont les travaux ont constitué une source d'inspiration majeure, et le Pr. Franck Iutzeler, qui a suivi mes avancées dès la deuxième année. J'adresse également mes remerciements au Pr. Michel De Lara et à la Dr. Delphine Bresch-Pietri pour leur participation au jury de soutenance et pour leurs remarques et corrections de ce manuscrit.

Je tiens à exprimer ma profonde reconnaissance à Paul Malisani, promoteur de cette thèse, pour son accompagnement constant, sa disponibilité et sa patience, ses conseils toujours avisés, sa bienveillance tout au long de ces années, et bien sûr pour les parties de babyfoot de haute voltige qui ont rythmé la vie à l'IFPEN.

Je tiens également à remercier Welington de Oliveira, directeur scientifique de cette thèse, pour son encadrement, sa rigueur et sa précision. Il m'a continuellement poussé à rechercher des contributions plus intéressantes et utiles, toujours avec le sourire et une bonne humeur communicative.

Je remercie Jiamin Zhu, co-promotrice de cette thèse, qui a toujours permis à l'équipe de prendre du recul sur les projets afin d'avancer dans la bonne direction.

Pour moi, le doctorat a été une belle aventure, rendue possible grâce à une équipe encadrante exceptionnelle : disponible, motivée, compétente, curieuse et pragmatique. Merci à vous trois.

Je remercie également l'IFPEN, une structure particulièrement agréable pour mener à bien une thèse, ainsi que tous mes collègues et co-doctorants de R11 et du CMA (Mines). J'adresse une pensée particulière à Sylvie, pour son aide précieuse dans de nombreuses démarches administratives, toujours accompagnée de son sourire et de sa bonne humeur.

Enfin, et surtout, je remercie tendrement mes parents. C'est grâce à votre éducation que j'ai développé cette curiosité pour la science, mais aussi pour tant d'autres domaines. Grâce à vous, j'ai mené mes études avec le désir d'aller toujours plus loin. Merci pour votre amour et pour le réconfort que je peux toujours trouver à la Maison.
Je tiens également à remercier mes frère et sœur, Elsa et Nathan, avec qui j'ai grandi et qui n'ont jamais cessé de me faire rire, c'est sûrement grâce à vous que j'aime autant sourire. Merci d'être toujours là.
Je vous réaffirme tout mon amour et vous dédie ces travaux.

Abstract

Decision making under uncertainty in multistage stochastic optimization poses a substantial challenge, necessitating a complex trade-off between, on the one hand, the representation of the uncertainties (i.e. the number of scenarios) and, on the other hand, the computational tractability. Scenario reduction methods, pioneered in 2003 by Dupačová et al. [52], offer a promising outlooks for achieving a satisfactory trade-off. However, the choice of distance metric for reducing scenario trees significantly influences solution quality. While clustering techniques have been prevalent, recent research has turned to Wasserstein-based methods to minimize transport distance between probabilty measures.

This work presents a comprehensive investigation of the use of the Wasserstein distance for scenario tree reduction in the context of multistage stochastic optimization. The Wasserstein Barycenter (WB) problem arises naturally in this setting: it serves as a tool for summarizing sets of probabilities, it appears in a number of disciplines, including applied probability, clustering and image processing. Numerically efficient methods to computing WBs rely on entropic regularization functions, resulting in approximate solutions due to limitations in solver capabilities. In contrast, this research introduces an exact approach based on the Douglas-Rachford splitting method directly applied to the WB linear optimization problem. The proposed solving algorithm achieves a trade-off between the numerical efficiency of regularization-based methods and the precision of exact LP solvers.

This thesis also proposes a novel formulation for the unbalanced optimal transport problem. This formulation leads to more flexible real-world applications than literature and the proposed WB method can be easily adapted to tackle unbalanced WB. Another new optimal transport problem is formulated in this thesis: the constrained WB. The motivations is to find WB laying on smaller support or to keep particular properties that can be very useful when performing scenario tree reduction. It also finds other applications in image processing or biomedical research for example. This work propose exact algorithms to handle this problem.

In [70], Kovacevic and Pichler develop an algorithm based on nested Wasserstein distance. As we point out in this work, this algorithm consists of computing a significant amount of Wasserstein barycenters. The third contribution of this work is to implement dedicated WB computation algorithms, including the Iterative Bregmann Projection method (IBP) and the newly introduced Method of Averaged Marginals (MAM) in the algorithm proposed in [70] to accelerate its performances.

The last contribution of this work is the development of a comprehensive benchmark comparing several optimization frameworks—including reinforcement learning, distributionally robust optimization, and stochastic programming—for solving a multistage energy management problem based on ground truth measurements of production and consumption of an IFPEN's building in Solaize. We integrate these approaches into a unified software pipeline that encompasses all key components developed in this thesis, including the scenario tree reduction algorithm enhanced by optimal transport techniques.

Résumé

La prise de décision sous incertitude dans les problèmes d'optimisation stochastique multi-étapes constitue un défi majeur, nécessitant un compromis complexe entre, d'une part, la représentation des incertitudes (c'est-à-dire le nombre de scénarios) et, d'autre part, la résolvabilité numérique. Les méthodes de réduction de scénarios, introduites en 2003 par Dupačová et al. [52], offrent une piste prometteuse pour atteindre ce compromis. Toutefois, le choix de la métrique de distance utilisée pour la réduction des arbres de scénarios a un impact significatif sur la qualité des solutions. Alors que les techniques de regroupement (clustering) sont couramment utilisées, des travaux récents se tournent vers des méthodes fondées sur la distance de Wasserstein, qui vise à minimiser la distance de transport entre mesures de probabilité.

Ce travail propose une étude approfondie de l'utilisation de la distance de Wasserstein pour la réduction d'arbres de scénarios dans le cadre de l'optimisation stochastique multi-étapes. Le problème du barycentre de Wasserstein (BW) y apparaît naturellement : il permet de résumer un ensemble de mesures de probabilité et intervient dans de nombreux domaines, tels que les probabilités appliquées, le regroupement de données et le traitement d'images. Les méthodes numériques efficaces pour calculer des BWs reposent souvent sur une régularisation entropique, fournissant des solutions approchées en raison des limitations des solveurs. En contraste, cette thèse introduit une méthode exacte fondée sur le schéma de décomposition de Douglas-Rachford, appliqué directement à la formulation linéaire du problème de BW. L'algorithme proposé constitue un compromis entre l'efficacité numérique des méthodes régularisées et la précision des solveurs linéaires exacts.

Cette thèse propose également une nouvelle formulation du problème de transport optimal déséquilibré. Cette formulation offre une plus grande flexibilité pour des applications concrètes que celles présentes dans la littérature, et la méthode proposée pour le barycentre de Wasserstein peut être facilement adaptée pour traiter les cas déséquilibrés. Par ailleurs, un autre problème nouveau de transport optimal est formulé dans cette thèse : le barycentre contraint (constrained WB). L'objectif est de trouver un barycentre supporté sur un ensemble plus restreint ou de conserver certaines propriétés spécifiques, ce qui s'avère particulièrement utile pour la réduction d'arbres de scénarios, ainsi que dans d'autres domaines tels que le traitement d'images. Des algorithmes exacts sont proposés pour résoudre ce problème.

Dans [70], Kovacevic et Pichler développent un algorithme de réduction fondé sur la distance de Wasserstein imbriquée. Comme nous le montrons dans cette thèse, cet algorithme repose sur le calcul d'un grand nombre de barycentres de Wasserstein. La troisième contribution de cette thèse consiste à intégrer des algorithmes dédiés au calcul des BW — notamment la méthode des projections de Bregman itérées (IBP) et la nouvelle méthode des moyennes marginales (MAM) — dans l'algorithme proposé par [70], afin d'en améliorer les performances.

La dernière contribution de ce travail est le développement d'une methode numérique complète (software) comparant plusieurs modèles d'optimisation — notamment l'apprentissage par renforcement, l'optimisation robuste de distribution et la programmation stochastique — pour la résolution d'un problème industriel de gestion de l'énergie multi-étapes, en utilisant des données mesurées de production et de consommation d'un bâtiment d'IFPEN à Solaize. Ces approches sont intégrées dans un logiciel unifié, regroupant l'ensemble des modules développés dans cette

thèse, dont l'algorithme de réduction d'arbres de scénarios enrichi par des techniques de transport optimal.

Contents

Chapter 1

Introduction

1.1 Context

The global energy landscape is transforming, driven by the imperative to reduce greenhouse gas emissions and mitigate climate change. Central to this transition is the increasing integration of renewable energy sources, such as wind and solar, into the electricity generation mix. While renewable energy offers significant environmental benefits, its inherent intermittency and variability present formidable challenges for grid operators and energy management systems.

In traditional electricity grids dominated by fossil fuels, power generation could be adjusted according to demand fluctuations, providing a stable and predictable supply. However, the rise of renewables introduces a new dynamic, where generation is subject to the variability of weather patterns and time-of-day fluctuations. This unpredictability disrupts the conventional energy supply and demand balance paradigm, necessitating innovative solutions for effective grid management.

One key strategy to address this challenge lies in enhancing flexibility on the demand side of the electricity grid. By enabling consumers to adjust their energy consumption patterns in response to real-time conditions, demand-side flexibility offers a powerful tool for optimizing the utilization of renewable energy resources. However, realizing this potential requires sophisticated Energy Management Systems (EMS) capable of orchestrating complex interactions between diverse stakeholders, energy assets, and market dynamics.

In this context, optimization-based energy management systems emerge as a promising approach to harness the full potential of renewable energy integration. By leveraging advanced mathematical techniques, such as stochastic optimization, these systems can intelligently allocate resources, optimize scheduling, and mitigate risks in the presence of uncertainty. Stochastic optimization techniques are well-suited to model and optimize complex, uncertain

systems, making them ideally suited for addressing the inherent variability of renewable energy generation.

1.2 Models for decision making under uncertainty

In this work we are considering multistage stochastic optimization problems:

$$\min_{\boldsymbol{u}_1} \mathbb{E}_{\boldsymbol{\xi}_1} \left[c_1(\boldsymbol{x}_1, \boldsymbol{u}_1, \boldsymbol{\xi}_1) + \min_{\boldsymbol{u}_2} \mathbb{E}_{\boldsymbol{\xi}_2} \left[c_2(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{\xi}_2) + \cdots + \min_{\boldsymbol{u}_T} \mathbb{E}_{\boldsymbol{\xi}_T} \left[c_T(\boldsymbol{x}_T, \boldsymbol{u}_T, \boldsymbol{\xi}_T) \right] \right] \right] \tag{1.1}$$

under the following constraints

$$\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_t) \qquad t = 1, \dots, T-1 \tag{1.2a}$$

$$(\boldsymbol{u}_t, \boldsymbol{x}_t) \in K_t \subset \mathbb{R}^m \times \mathbb{R}^n \qquad t = 1, \dots, T \tag{1.2b}$$

$$x_1 = x^0, \tag{1.2c}$$

Bold characters are used to denote random variables. We denote $\boldsymbol{u}_t \in \mathbb{R}^m$ the decision vector at stage t, $\boldsymbol{x}_t \in \mathbb{R}^n$ the state variable, $\boldsymbol{\xi}_t \in \mathbb{R}^p$ denotes the random vector at stage t, $c_t : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ denotes the cost function at stage t, f_t is the system dynamics and K_t is a closed convex set. From the recursive structure of the objective function, one can deduce that an optimal decision at stage t, denoted $\bar{\boldsymbol{u}}_t$, depends on the realizations of the past random variables denoted $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)$ and on the statistical properties of the future random variables $(\boldsymbol{\xi}_{t+1}, \dots, \boldsymbol{\xi}_T)$. Thus, an optimal solution of eq. (1.1) is said to be *non-anticipative*: at stage t, we have decided $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_t)$, only using the information available at each stage.
Solving the problem from (1.1) and (1.2) in its full generality is a complex task, and thus multiple optimization models have been developed. The reader can refer to Shapiro et al. [111] and Pflug and Pichler [95]. The following section provides an overview of these methods.

1.2.1 MPC for multistage stochastic optimal control problems

At each time step t, Model Predictive Control (MPC) algorithms [58, 71, 88] consist in predicting a realization of the next K random variables denoted $(\hat{\xi}_t, \dots \hat{\xi}_{t+K})$, and solve the following deterministic optimization problem

$$\min_{u_t, \dots, u_{t+K}} \sum_{s=t}^{t+K} c_s(x_s, u_s, \hat{\xi}_s) \tag{1.3}$$

under constraints described in eq. (1.2), and keep u_t as the decision.
These methods are easy to implement and rely on deterministic optimization algorithms. However, the optimization problem from (MPC) does not use the full statistical properties of the future random variables, potentially yielding far from sub-optimal decisions with respect to the original optimization problem from eqs. (1.1) and (1.2). MPC methods rely on a single forecast of the uncertain variables, and thus depend heavily on the operator's confidence in the prediction model. However, in practice, such forecasts can significantly deviate from actual outcomes.

1.2.2 Dynamic programming-based optimization methods

Stochastic optimization methods are often based on sequential decomposition methods, which rely on the principle of dynamic programming [11,13,14,16]. Recently, significant work has been done on Stochastic Dual Dynamic Programming methods [20,28,89,109,112]. These methods have the advantage of naturally computing non-anticipative decisions and of being able to take into account a large number of uncertainties scenarios. On the other hand, these methods require strong the assumption of stage-wise independence. In the context of energy management problems, this assumption is not always satisfied.

1.2.3 Scenario decomposition for stochastic optimal control problems

Scenario decomposition techniques use a different paradigm for stochastic optimization problems: instead of stage decomposition, i.e. minimizing a recursive sequence of recourse functions, scenario decomposition techniques decompose the problem per scenario while keeping the whole time horizon in individual (scenario-based) subproblems. This different paradigm requires a different manner to deal with nonanticipativity, which strongly depends on the scenarios. Therefore, scenario decomposition techniques consist in optimizing over a set of functions defined on a discrete probability space denoted $(\Omega, \mathcal{F}, P)$. The function spaces on this probability space are denoted as follows

$$\mathbb{U} := \{\boldsymbol{u} : \Omega \mapsto \ell_2([1,\dots,T];\mathbb{R}^m)\}, \tag{1.4}$$

$$\mathbb{X} := \{\boldsymbol{x} : \Omega \mapsto \ell_\infty([1,\dots,T];\mathbb{R}^n)\}, \tag{1.5}$$

$$\Xi := \left\{\boldsymbol{\xi} : \Omega \mapsto \ell_2([1,\dots,T];\mathbb{R}^d)\right\}. \tag{1.6}$$

Given $p \in [1,+\infty]$, we denote $\ell_p([1,\dots,T];\mathbb{R}^m)$ the space of p-summable sequences of length T valued in $\mathbb{R}^m$, and we denote $\|.\|_{\ell_p}$ the corresponding p-norm. We endow the space ℓ_2 with the standard inner product

$$\langle x, y\rangle_{\ell_2} := \sum_{t=1}^{T} x_t^\top y_t \tag{1.7}$$

We endow the space $\mathbb{U}$ with the following inner product

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathbb{U}} := \mathbb{E}_P\left[\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\ell_2}\right] \tag{1.8}$$

and we denote $\|.\|_{\mathbb{U}}$ its associated norm.

1.2.3.1 Stochastic Optimization models

We now present several well-known modeling approaches solving stochastic optimization problems. The key challenge is to identify which formulation is best suited to the multistage stochastic optimization problem we will study.

Definition 1. *(Stochastic Programming model – SP) The multistage optimization problem can be formulated as*

$$\inf_{\boldsymbol{u}\in\mathbb{U}} \left\{ \mathbb{E}_P\left[F(\boldsymbol{u},\boldsymbol{\xi})\right] := \mathbb{E}_P\left[\sum_{t=1}^{T} c_t(\boldsymbol{x}_t[\boldsymbol{u}], \boldsymbol{u}_t, \boldsymbol{\xi}_t)\right] \right\} \tag{SP}$$

under the following constraints

$$\boldsymbol{x}_{t+1}[\boldsymbol{u}] = f_t(\boldsymbol{x}_t[\boldsymbol{u}], \boldsymbol{u}_t, \boldsymbol{\xi}_t) \quad P-a.s., \tag{1.9a}$$
$$\boldsymbol{x}_1 = x^0 \quad P-a.s., \tag{1.9b}$$
$$(\boldsymbol{x}_t[\boldsymbol{u}], \boldsymbol{u}_t) \in K_t \quad P-a.s., \tag{1.9c}$$
$$\boldsymbol{u} - \mathbb{E}_P[\boldsymbol{u}|\boldsymbol{\xi}] = 0 \quad P-a.s. \tag{1.9d}$$

Where $\boldsymbol{x}[\boldsymbol{u}]$ is the unique solution of eqs. (1.9a) and (1.9b). The notation $\mathbb{E}_P[\boldsymbol{u}|\boldsymbol{\xi}]$ is the projection, with respect to $\|.\|_{\mathbb{U}}$, onto the subspace of functions adapted to $\boldsymbol{\xi}$, i.e., it is the projection on the non-anticipative space. Therefore, constraint (1.9d) ensures that the control $\boldsymbol{u}$ is non-anticipative.
For notational convenience, we define

$$\mathbb{C} := \{\boldsymbol{u} \in \mathbb{U} : (1.9) \text{ holds}\}. \tag{1.10}$$

The stochastic programming model introduced in (SP) minimizes expected costs by anticipating among a wide range of future scenarios. This model assumes that the probability distribution P of future data is known in advance. Although reasonable, this assumption is not always satisfied in practice, and the true distribution sometimes has to be estimated from limited historical data, which may affect the reliability of the results derived from it.

Definition 2. *(Robust Optimization model – RO)*

$$\inf_{\boldsymbol{u}\in\mathbb{C}} \max_{\boldsymbol{\xi}\in\Xi} F(\boldsymbol{u}, \boldsymbol{\xi}). \tag{RO}$$

Robust Optimization (RO) [55,116] discards the need for probabilities altogether. Instead, it assumes the worst-case scenario among all the possible outcomes. This approach yields conservative solutions, which are often too costly or overly cautious for practical use [59].

A more balanced approach, known as Distributionally Robust Optimization (DRO), considers distributions lying within an ambiguity set $\mathcal{P}$ — a set of plausible probability distributions constructed from a finite collection of scenarios. The goal is to optimize decisions to perform well against the worst-case distribution in this set.

Definition 3. *(Distributionally Robust Optimization model – DRO)*

$$\inf_{\boldsymbol{u}\in\mathbb{C}} \sup_{P\in\mathcal{P}} \mathbb{E}_P[F(\boldsymbol{u}, \boldsymbol{\xi})] \tag{DRO}$$

In fact, rather than relying on a single estimated probability measure, DRO acknowledges the uncertainty in this estimation and instead considers all distributions that lie within a certain neighborhood of the empirical measure. The quality of the DRO solution strongly depends on how this ambiguity set is defined as it captures the trade-off between robustness and reliance on a specific distribution.

Definition 4. *(Variance Penalized Stochastic Programming model – VSP)*

$$\inf_{\boldsymbol{u}\in\mathbb{C}} \mathbb{E}_P\left[[F(\boldsymbol{u}, \boldsymbol{\xi})] + \frac{\alpha}{2}\|\boldsymbol{u} - \mathbb{E}_P(\boldsymbol{u})\|_{\ell_2}^2\right] \tag{VSP}$$

The recent work [80], situated within the field of risk-averse stochastic optimization, introduces a novel approach that integrates variance penalization directly into multistage SP models. By accounting for cost variability across scenarios, the approach effectively balances robustness and performance.

Another approach is the deterministic-control approach to stochastic programming (DSP) model. It involves computing a single control policy that minimizes the expected cost across all scenarios.

Definition 5. *(Desterministic-Control Approach for SP – DSP)*

$$\inf_{u\in\ell_2([1,\dots,T];\mathbb{R}^n)} \mathbb{E}_P[F(u,\boldsymbol{\xi})] \quad \textit{so that eqs. (1.9a) to (1.9c) hold} \tag{DSP}$$

Note that the control u here does not depends on $\boldsymbol{\xi}$.
(DSP) fits a standard setting of constrained optimal control problem and can be solved easily even when using a large number of scenarios. This method is highly robust but can be sub-optimal. Also note that this model can have no solution because it is possible that there exists no deterministic control that can respect the constraints for all scenarios.

A **key challenge** shared by most of the optimization models discussed so far is their reliance on scenario generation. While increasing the number of scenarios generally improves solution accuracy, it also significantly increases computational complexity. To enable a fair and rigorous comparison between models (against DSP for which the number of scenarios is not an issue), it is essential that each problem be solved efficiently by suitable methods under their best performance conditions. One of the contributions of this work is to maintain tractability of stochastic optimization models while leveraging a large number of scenarios.

1.3 Scenario reduction for multistage stochastic optimization problem

1.3.1 From scenario decomposition to scenario trees

At each time step t, the random variables $\boldsymbol{\xi}_{[t]} := (\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_t)$ generate a $\tilde{\mathcal{A}}_t$ partition of Ω defined as

$$\tilde{\mathcal{A}}_t := \{A \subseteq \Omega : \forall \omega_1, \omega_2 \in A, \forall s \leq t, \boldsymbol{\xi}_s(\omega_1) = \boldsymbol{\xi}_s(\omega_2)\}\,. \tag{1.11}$$

The elements of the partition $\tilde{\mathcal{A}}_t$ are called t-atoms and we denote $\mathcal{F}_t$ the smallest σ-algebra generated by the t-atoms, and the sequence $(\mathcal{F}_t)_t$ is the filtration generated by the random variables $\boldsymbol{\xi}_{[t]}$. Finally the sequence of t-partition can be represented as a scenario tree, where each node of depth t corresponds to a t-atom as illustrated on Figure 1.1. The non-anticipative constraint (1.9d) ensures that the optimal solution $\boldsymbol{u}$ yields the same t-partition of Ω as $\boldsymbol{\xi}$ and thus yields the same filtration $(\mathcal{F}_t)_t$. Now, using this t-partition computing the conditional expectation $\boldsymbol{u} := \mathbb{E}[\hat{\boldsymbol{u}}|\boldsymbol{\xi}]$ is straightforward and we have

$$\forall A \in \tilde{\mathcal{A}}_t, \forall \omega \in A, \boldsymbol{u}_{[t]}(\omega) := \frac{1}{P(A)} \sum_{\omega' \in A} P(\{\omega'\})\, \hat{\boldsymbol{u}}_{[t]}(\omega'). \tag{1.12}$$

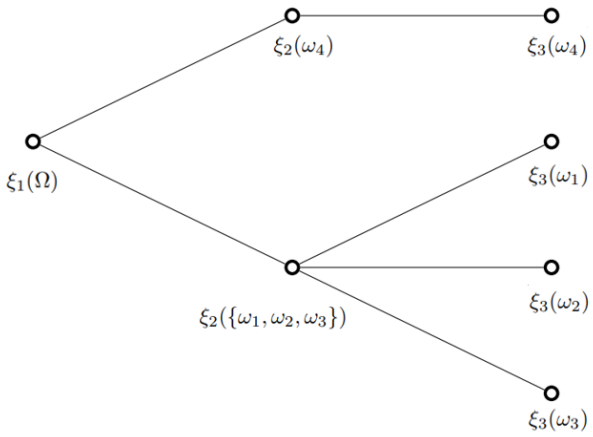


Figure 1.1: Let $\Omega = \{\omega_1, \ldots, \omega_4\}$. At stage $t = 1$, all scenarios have the same value, i.e., $\boldsymbol{\xi}_1(\omega_i) = \boldsymbol{\xi}_1(\omega_j)$, for all i, j. At stage $t = 2$, scenarios $\{\omega_1, \omega_2, \omega_3\}$ are still identical whereas $\boldsymbol{\xi}_2(\omega_4) \neq \boldsymbol{\xi}_2(\{\omega_1, \omega_2, \omega_3\})$. Finally, at stage $t = 3$, all scenarios are different, $\boldsymbol{\xi}_3(\omega_i) \neq \boldsymbol{\xi}_3(\omega_j)$, for all $i \neq j$.

1.3.2 The scenario tree reduction problem

As illustrated in Section 1.3.1, the random variables and the available information at stages $t = 1, \ldots, T$ in a stochastic optimization problem with finitely many scenarios can be represented as a scenario tree. The complexity of the multistage stochastic optimization problem increases with the number of scenarios. Therefore the necessity of finding an equilibrium between the representation of uncertainties and numerical tractability is strongly linked to the concept of scenario tree reduction. This concept was introduced by Dupačová et al. [52] in 2003. The main idea is to start from a large scenario tree that accurately represents the underlying stochastic process, and to construct a smaller tree that preserves as much of the original information as possible. A fundamental question that arises is how to formally quantify the information preservation.

When reducing scenario trees, the choice of a distance to minimize between trees is of primary importance. There exist distances, and thus methods, that ignore the accumulation of information over time on which rely the non-anticipativity of stochastic problems.

Notably, one of the most commonly used scenario reduction algorithms is the approach proposed by Heitsch and Römisch [65], which measures distances between nodes with the same parent, ultimately reducing pairs of nodes until a stopping criterion is met. The work [128] developed algorithms that employ K-means clustering and Linear Programming (LP) moment-matching methods to approximate multistage scenario trees. In [67] comparative analysis of several clustering algorithms are conducted, and highlight that the general behaviour of these algorithms concerning the closeness to the exact solution of the stochastic problem is still volatile.
Wasserstein distance-based methods aim to minimize probabilistic distances, akin to the Wasserstein distance [105, 122], between the original complex tree and a simpler smaller tree whose solution computation will be more tractable; see for instance [46,52]. Li and Floudas [74] treat scenario reduction as a mixed-integer linear programming

(MILP) problem, and aim at reducing the Wasserstein distance between scenarios. Subsequent works [69, 75] improved other LP-based scenario reduction strategies, the latter solved the entropy-regularized optimal transport using the Sinkhorn–Knopp algorithm [115].

The Wasserstein distance approach is a natural choice when tackling probability distributions but does not capture the structural information of multistage scenario trees. Neglecting the tree filtration potentially leads to deviations in solutions, because the solution of a stochastic optimization problem is non-anticipative. This is why it is of great importance to use a reduction method that takes into account the tree structure, i.e. the *filtration*. Indeed, the authors of [66] studied the stability of multistage stochastic programs and showed that multistage scenario tree reductions should be based on a distance that accounts for filtration in order to obtain a bound on the stability of the solution.

The introduction of the *nested distance* [96, 97], also called *process distance*, offered a valuable framework for stochastic programs. This new way of computing a distance between trees that takes into account the filtration, is now broadly used to compare trees and assess the accuracy of reduction methods. Leveraging the nested distance to guide the process approximation enables control over both the statistical quality of the approximation and its impact on the objective in multistage stochastic optimization problems pertaining to the respective stochastic process [66,70]. Kovacevic and Pichler [70] introduced an algorithm that directly targets the minimization of the nested distance in the reduction tree process, although this approach is proved complex and computationally expensive. Indeed while it is easy to calculate the process distance between given trees by solving one LP, finding a tree to minimize the process distance (i.e. reducing a tree) is much more difficult. Both probabilities and values of the approximating tree have to be chosen, such that the process distance is minimized. This leads to a large, nonconvex optimization problem, which can be solved in reasonable time only for small instances due to the computation of multiple potentially large-scale LPs.

Based on the Kovacevic and Pichler's work [70], we introduce a novel approach for reducing general scenario trees. We noticed that the minimization of the nested distance between two trees, as employed in the Kovacevic and Pichler's method, makes naturally appear the Wasserstein barycenter problem. We propose to leverage this optimal transport problem and employ efficient algorithms to attractively enhance the Kovacevic and Pichler's reduction method.

In this context, we developed the Method of Averaged Marginals (MAM) [85] as a novel and efficient method for computing Wasserstein barycenters. Unlike iterative projection-based methods, such as the Iterative Bregman Projection algorithm (IBP) [15] — a widely used state-of-the-art approach — MAM is grounded in a marginal-based formulation that enables direct optimization without entropic regularization. We provide a detailed convergence analysis of the method, propose accelerated variants, and demonstrate its practical relevance through extensive numerical experiments on real-world data.

1.4 The Energy Management System battery strategy at IFPEN

This work takes its motivations in the energy managing challenges raised by the increasing integration of renewable energy sources into the power grid.

An **Energy Management System** (EMS) is a software system — sometimes coupled with physical controllers — designed to optimize energy usage within a building, industrial site, power network, or microgrid. Its purpose is to monitor and control energy flows in order to reduce operational costs, improve efficiency, and support integration of local renewable sources and storage devices. Depending on the application, the EMS may act on various time scales (real-time, hourly, daily) and control various components (batteries, loads, production units) through automatic strategies.

In our setting, the EMS is in charge of the automatic control of a stationary battery connected downstream of a prosumer's electricity meter. A *prosumer* is an end-user which both consumes and produces electricity, typically via intermittent sources such as photovoltaic panels, as depicted Figure 1.2. The goal is to minimize the expected electricity cost over a finite time horizon, while accounting for battery dynamics and the stochastic nature of both **consumption** and **production**.

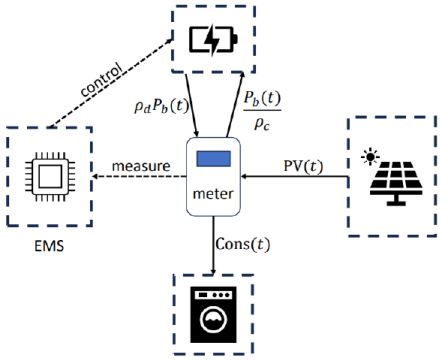


Figure 1.2: Schematic diagram of a domestic system with a stationary battery controlled by an EMS [80]

However, managing energy efficiently requires making sequential decisions under uncertainty: the future is not fully known when decisions must be made. In this context, the methods developed in this thesis aim to address this challenge by computing robust models, which naturally involve large scenario trees—hence the need for scenario tree reduction techniques.

1.5 Contributions and outline

This thesis presents both methodological and applied contributions in the field of energy management optimization under uncertainty. All proposed methods are applied and benchmarked on a real EMS deployed at IFPEN. Our main contributions can be summarized as follows:

Chapter 2 addresses the core optimization problem underlying the scenario tree reduction process. We first review key concepts from optimal transport theory, including the Wasserstein distance and its role in averaging probability distributions. We then introduce

- **a new algorithm for the Wasserstein barycenter problem (WB).** We develop a novel optimization algorithm for solving the Wasserstein barycenter (WB) problem exactly: the *Method of Averaged Marginals* (MAM).
  - The algorithm is based on the Douglas–Rachford operator splitting scheme. We show that each step of the splitting procedure is computationally tractable;
  - we provide mathematical guarantees of exact convergence based on the Douglas-Rachford theory in the deterministic and randomized version of the algorithm;
  - we benchmark MAM against state-of-the-art methods such as Iterative Bregman Projections (IBP) [15] and show the strenght of MAM;
  - we introduce a new model for the *unbalanced* Wasserstein barycenter problem and demonstrate its practical applications through numerical experiments;
  - we develop mathematical garantee for MAM from fixed-support to free-support settings.

  The main content of Chapter 2 appears in the paper: Mimouni, Malisani, Zhu, de Oliveira, (2024) *Computing Wasserstein barycenters via operator splitting: The method of averaged marginals.* SIAM Journal on Mathematics of Data Science, 6(4), 1000-1026 [85].

Chapter 3 extends the classical barycenter problem by incorporating convex and nonconvex constraints on the solution. These constrained formulations are motivated by real-world applications such as scenario tree reduction, where the barycenter can be requested to satisfy structural or physical constraints. We propose

- **a new model and approach for computing constrained WBs.**
  - We propose several extensions of the WB formulation incorporating convex and nonconvex constraints on the barycenter. These constraints are particularly relevant in the context of tree reduction and other industrial use cases;
  - we show that a slight modification of MAM solves the problem exactly in the case of convex constraints;
  - we propose an easy-to-implement heuristic for solving the problem with nonconvex constraints, and we motivate its use through examples in which it performs well;
  - we propose a general exact method to solve the problem if the constraints are nonconvex.

  The main content of Chapter 3 will appear in the paper: Mimouni, de Oliveira, Sempere, (2025). *On the Computation of Constrained Wasserstein Barycenters.* Pacific Journal of Optimization for a special issue dedicated to Professor R. Tyrrell Rockafellar [84].

Chapter 4 presents our methodology for scenario tree reduction. After introducing the notion of nested distance between stochastic processes, we detail how the reduction task can be reformulated as a sequence of Wasserstein barycenter problems. We describe the algorithmic implementation of our approach and compare the original method of Kovacevic and Pichler with the enhanced versions proposed in this work through extensive numerical experiments. We propose

- **a boosted scenario tree reduction methodology.** Increasing the number of scenarios improves model fidelity but leads to significant computational burden. To tackle this, we:

  - build on the work of Kovacevic and Pichler [70], which involves minimizing the nested distance between scenario trees;
  - demonstrate that this approach naturally leads to solving a sequence of Wasserstein barycenter problems;
  - enhance their method using modern optimal transport algorithms, including IBP and our newly proposed MAM;
  - achieve up to a 9-fold improvement in CPU time for large trees, based on our numerical experiments.

  These results offer a practical and scalable tree reduction strategy for complex stochastic systems. The main content of Chapter 4 have been submitted for possible publication in: Mimouni, Malisani, Zhu, de Oliveira, (2025) *Scenario Tree Reduction via Wasserstein Barycenters.* Annals of Operations Research [86].

Chapter 5 turns to the EMS control application at IFPEN. We compare multiple decision-making models. This chapter provides practical insights into the trade-offs between model complexity, robustness, and computational efficiency, and includes a complete implementation pipeline adapted to IFPEN's industrial system. We conduct

- **a comprehensive comparison of decision-making models.** This includes MPC, (risk-free and risk-averse) stochastic programming models, robust optimization, distributionally robust models, and reinforcement learning. We highlight that:
  - scenario-based models can be effectively deployed thanks to scenario reduction techniques that preserve as much of the original information as possible;
  - each model is evaluated in terms of robustness, computational tractability, and out-of-sample performance on real data from the IFPEN battery;
  - we develop a complete end-to-end software pipeline tailored to IFPEN's system architecture, integrating scenario generation, tree reduction, optimization, and automatic control execution.

  The main content of Chapter 5 will be submitted to a reputable journal in the field of EMS [87].

Chapter 2

The Wasserstein barycenter problem

This chapter addresses the core optimization problem underlying the scenario tree reduction process.

Abstract The Wasserstein barycenter (WB) is an important tool for summarizing sets of probability measures. It finds applications in applied probability, clustering, image processing, etc. When the measures' supports are finite, computing a (balanced) WB can be done by solving a linear optimization problem whose dimensions generally exceed standard solvers' capabilities. In the more general setting where measures have different total masses, we propose a convex nonsmooth optimization formulation for the so-called unbalanced WB problem. Due to their colossal dimensions, we introduce a decomposition scheme based on the Douglas-Rachford splitting method that can be applied to both balanced and unbalanced WB problem variants. Our algorithm, which has the interesting interpretation of being built upon averaging marginals, operates a series of simple (and exact) projections that can be parallelized and even randomized, making it suitable for large-scale datasets. Numerical comparisons against state-of-the-art methods on several data sets from the literature illustrate the method's performance.

The main content of this chapter has appeared in [85]: Mimouni, Malisani, Zhu, de Oliveira, (2024) *Computing Wasserstein barycenters via operator splitting: The method of averaged marginals.* SIAM Journal on Mathematics of Data Science, 6(4), 1000-1026.

2.1 Introduction

In applied probability, stochastic optimization, and data science, a crucial aspect is the ability to compare, summarize, and reduce the dimensionality of empirical/discrete measures. Since these tasks rely heavily on pairwise comparisons of measures, it is essential to use an appropriate metric for accurate data analysis. Different metrics define different barycenters of a set of measures: a barycenter is a mean element that minimizes the (weighted) sum of all its square distances to the set of target measures. When the chosen metric is the optimal transport one,

and there is mass equality between the measures, the underlying barycenter is denoted by (balanced) Wasserstein Barycenter (WB).

The optimal transport metric defines the so-called Wasserstein distance (also known as Mallows or Earth Mover's distance), a popular choice in statistics, machine learning, and stochastic optimization [47,93,95]. The Wasserstein distance has several valuable theoretical and practical properties [105,122] that are transferred to WBs [1,38,93,99]. Indeed, thanks to the Wasserstein distance, one key advantage of WBs is their ability to preserve the underlying geometry of the data, even in high-dimensional spaces. This fact makes WBs particularly useful in image processing, where datasets often contain many pixels and complex features that must be accurately represented and analyzed [113, 118].

Being defined by the Wasserstein distance, WBs are challenging to compute. The Wasserstein distance is computationally expensive because, to compute an optimal transport plan, one needs to cope with a large linear program (LP) problem that has no analytical solution and cubic worst-case complexity[1] [131]. The situation becomes even worse for computing a WB of a set of finitely many discrete measures as the problem involves several transport plans [1]. This problem can be written as LP [6, 27], whose size becomes astronomical as it scales exponentially in the number of measures, exceeding thus the capabilities of standard LP solvers even for a small number of measures [6, 23, 27]. For this reason, significant effort has been made to reduce the LP's size and design specialized solvers [2, 6, 22, 27]. In particular, the work [22] proposes reduced LP models that exploit data structure. Although significantly smaller than the original LP problem defining WBs, those models are in general large scale and still hard to solve. The work [2] leverages techniques from computational geometry and combinatorial optimization to propose a specialized LP solver for computing WBs. The approach, which works on the dual problem and implements a separation oracle, is not efficient beyond moderate-scale inputs [2, § 5]. Indeed, a WB cannot be computed in time polynomial in the number of measures, (maximum) support size, and dimension [3].

Given the difficulty of computing exact (free-support) WBs, much research has focused on inexact approaches. A vast body of literature focuses on computing inexact WBs, either by employing approximate LP approaches as in [21, 99, 123], or by restricting the support of the WB to a fixed set, the so-called fixed-support approaches [38,93,131]. These techniques often employ a block-coordinate scheme consisting of two steps, first fixing the support and optimizing over the masses, then fixing the masses and optimizing over the support (of a given size). The first of these steps is an LP problem with the same structure as the exact (free-support) WB's LP formulation discussed above. The only difference is the LP's size, as fixing the support reduces the problem significantly. The second step in the block-coordinate scheme has a straightforward solution, provided the quadratic Wasserstein distance is employed.

Hence, whether an exact or inexact approach is employed to compute (approximate) a WB, one invariably has to face a large-scale LP of the form (see equations (2.9) and (2.10) for details)

$$\min_{\pi \in \mathcal{B}} \sum_{m=1}^{M} \langle c^{(m)}, \pi^{(m)} \rangle \quad \text{s.t.} \quad \pi^{(m)} \in \Pi^{(m)}, \quad m = 1, \ldots, M, \tag{2.1}$$

where M stands for the number of discrete measures, c for the transportation costs, $\Pi^{(m)}$ represents a polytope containing measures with given marginal, and $\mathcal{B}$ symbolizes a linear subspace. While exact techniques usually build upon linear programming techniques, inexact approaches tackle (2.1) via reformulations based on an entropic regularization [15, 38, 40, 60, 93, 131]. Indeed, the work [38] proposes to compute a WB inexactly by decomposing

[1] More precisely, $O(S^3 log(S))$, with S the size of the input data.

(2.1) along the measures and then regularizing the resulting optimal transportation problems with an entropy-like function. A projected subgradient method gives rise to a minimization scheme with decomposition to deal with the high dimensions of the LP. The regularization technique allows one to employ the celebrated Sinkhorn algorithm [37,114], which has a simple closed-form expression and can be implemented efficiently using only matrix operations. Furthermore, this technique opened the way to the *Iterative Bregman Projection* (IBP) method proposed in [15]. IBP is highly memory efficient for distributions with a shared support set and is considered to be one of the most effective methods to tackle fixed-support WB problems. However, as IBP works with an approximating model and fixed support, the method falls in the class of inexact approaches.

Another approach fitting into the category of inexact methods has been recently proposed in [131], which uses the same type of regularization as IBP but decomposes the problem into a sequence of smaller subproblems with straightforward solutions. More specifically, the approach in [131] is a modification (tailored to the WB problem) of the *Bregman Alternating Direction Method of Multipliers* (B-ADMM) of [124]. The modified B-ADMM has been shown to compute promising results for sparse support measures and therefore is well-suited in some clustering applications. However, the theoretical convergence properties of the modified B-ADMM algorithm are not well understood and the approach should be considered as a heuristic. In the same vein, the work [130] proposes to address the WB problem via the standard ADMM algorithm, which decomposes the problem into smaller and simpler subproblems. As mentioned by the authors in their subsequent paper [131], the numerical efficiency of the standard ADMM is still inadequate for large datasets.

To cope with the challenge of solving LPs of the form (2.1) resulting from computing exact (free-support) or inexact (fixed-support) WBs, we propose a new algorithm based on the celebrated Douglas-Rachford splitting operator method (DR) [50, 53, 56]. Our proposal, which exploits the problem structure for decomposition, is denoted by *Method of Averaged Marginals* (MAM) as at every iteration, the algorithm computes a barycenter approximation by averaging marginals issued by transportation plans that are updated independently, in parallel, and even randomly if necessary. Accordingly, the algorithm operates a series of simple and exact projections that can be carried out in parallel and even randomly. These compelling features allow for considering data sets beyond moderate sizes in the free-support setting and attaining more accurate results than entropy-based methods usually get in the fixed-support case. Furthermore, MAM can be applied to a more general setting where measures have different total masses.

All the methods mentioned in the above references deal exclusively with sets of probability measures because WBs are limited to measures with equal total masses. A tentative way to circumvent this limitation is to normalize general positive measures to compute a standard (balanced) WB. However, such a naive strategy is generally unsatisfactory and limits the use of WBs in many real-life applications such as logistics, medical imaging, and others coming from the field of biology [63, 107]. Consequently, the concept of WB has been generalized to summarize such more general measures. Different generalizations of the WB exist in the literature, and they are based on variants of *unbalanced optimal transport problems* that define a distance between general non-negative, finitely supported measures by allowing for mass creation and destruction [63]. Essentially, such generalizations, known as unbalanced Wasserstein barycenters (UWBs), depend on how one chooses to relax the marginal constraints. In the review paper [107] and references therein, marginal constraints are moved to the objective function with the help of divergence functions. Differently, in [63] the authors replace the marginal constraints with sub-couplings and penalize their discrepancies. It is worth mentioning that UWB is more than simply copying with global variation in the measures' total masses. Generalized barycenters tend to be more robust to local mass variations, which include outliers and missing parts [107].

For the sake of a unified algorithmic proposal for both balanced and unbalanced WBs, in this work, we consider a different formulation for dealing with sets of measures with different total masses. Instead of relaxing both marginal constraints in each one of the M transportation plans as done in [63, 107] and references therein, our formulation generalizes the balanced WB by relaxing the constraint that the barycenter is a marginal measure of all underlying transportation plans. More specifically, by using the distance function to the subspace $\mathcal{B}$, that is $\texttt{dist}_{\mathcal{B}}(\pi) := \min_{\theta\in\mathcal{B}} \|\theta - \pi\|$, and a penalty parameter $\gamma > 0$, we propose the following nonlinear optimization problem yielding a UWB:

$$\min_{\pi} \sum_{m=1}^{M} \langle c^{(m)}, \pi^{(m)} \rangle + \gamma \texttt{dist}_{\mathcal{B}}(\pi) \quad \text{s.t.} \quad \pi^{(m)} \in \Pi^{(m)}, \quad m = 1, \ldots, M. \tag{2.2}$$

While our approach can be seen as an abridged alternative to the thorough methodologies of [63] and [107], its favorable structure for efficient splitting techniques combined with the good quality of the issued UWBs confirms the formulation's practical interest.

Thanks to our unified analysis, MAM can be applied to both balanced and unbalanced WB problems without any change: all that is needed is to set up the parameter $\gamma > 0$ in (2.2). To the best of our knowledge, MAM is the first approach capable of handling balanced and unbalanced WB problems in a single algorithm, which can be further run in a deterministic or randomized fashion. In addition to its versatility, MAM copes with scalability issues arising from barycenter problems, is memory efficient, and has convergence guarantees. As further contributions, we conduct experiments on several data sets from the literature to demonstrate the computational efficiency and accuracy of the new algorithm and make our Python codes publicly available at the link (`https://github.com/dan-mim/Computing-Wasserstein-Barycenters-MAM`) [81].

The remainder of this work is organized as follows. Section 2.2 introduces the notation and recalls the formulation of balanced WB problems. The proposed formulation for unbalanced WBs is presented in Section 2.3. Section 2.4 briefly recalls the Douglas-Rachford splitting (DR) method and its convergence properties both in the deterministic and randomized settings. The main contribution of this work, the Method of Averaged Marginals, is presented in Section 2.5. Convergence analysis is given in the same section by relying on the DR algorithm's properties. Section 2.6 illustrates the numerical performance of the deterministic and randomized variants of MAM on several data sets from the literature. Numerical comparisons with the free-support method [2] and fixed-support approaches in [15] and [131] are presented for the balanced case. Then, some applications of the UWB are considered.

2.2 Background on optimal transport and Wasserstein barycenters

Throughout this work, for $\tau \geq 0$ a given scalar, the notation $\Delta_R(\tau)$ denotes the set of vectors in $\mathbb{R}^R_+$ adding up to τ, that is,

$$\Delta_R(\tau) := \left\{ u \in \mathbb{R}^R_+ : \sum_{i=1}^{R} u_i = \tau \right\}. \tag{2.3}$$

If $\tau = 1$, then $\Delta_R(1)$, denoted simply by Δ_R, is the $R+1$ simplex. Let $\mathcal{P}(\mathbb{R}^d)$ be the set of Borel probability measures on $\mathbb{R}^d$. Furthermore, let ξ and ζ be two random vectors having probability measures μ and ν in $\mathcal{P}(\mathbb{R}^d)$,

that is, $\xi \sim \mu$ and $\zeta \sim \nu$. Their (quadratic) 2-Wasserstein distance is given by:

$$W_2(\mu,\nu) := \left(\inf_{\pi \in U(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\xi - \zeta\|^2 \mathrm{d}\pi(\xi,\zeta) \right)^{1/2}, \tag{WD}$$

where $U(\mu,\nu)$ is the set of all probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ having marginals μ and ν. We denote by $W_2^2(\mu,\nu)$ the squared Wassserstein distance, i.e., $W_2^2(\mu,\nu) := (W_2(\mu,\nu))^2$.

Definition 6 (Wasserstein Barycenter - WB)**.** *Given M measures $\{\nu^{(1)},\dots,\nu^{(M)}\}$ in $\mathcal{P}(\mathbb{R}^d)$ and $\alpha \in \Delta_M$, a* Wasserstein barycenter *with weights α is a solution to the following optimization problem*

$$\min_{\mu \in \mathcal{P}(\mathbb{R}^d)} \sum_{m=1}^{M} \alpha_m W_2^2(\mu,\nu^{(m)}) \,. \tag{2.4}$$

Informally, a WB μ is a measure such that the total cost for transporting from μ to all $\nu^{(m)}$ is minimal concerning the quadratic Wasserstein distance. A WB μ exists in generality and, if one of the $\nu^{(m)}$ vanishes on all Borel subsets of Hausdorff dimension $d-1$, then it is also unique [1]. In this work, we are given M empirical (discrete) measures $\nu^{(m)}$ having finite support sets:

$$\mathtt{supp}(\nu^{(m)}) := \left\{ \zeta_1^{(m)},\dots,\zeta_{S^{(m)}}^{(m)} \right\} \quad \text{and} \quad \nu^{(m)} = \sum_{s=1}^{S^{(m)}} q_s^{(m)} \delta_{\zeta^{(m)}}, \tag{2.5}$$

with δ_u the Dirac unit mass on $u \in \mathbb{R}^d$ and $q^{(m)} \in \Delta_{S^{(m)}}$, $m = 1,\dots,M$. In this case, the uniqueness of WB is no longer ensured in general but the following results hold [6].

Theorem 1 (From [6])**.** *Consider M empirical measures $\nu^{(m)}$ as in (2.5). Then, problem (2.4) has at least one solution.*

a) Every solution μ satisfies

$$\mathtt{supp}(\mu) \subset \Xi := \left\{ \sum_{m=1}^{M} \alpha_m \zeta_s^{(m)} \,:\, \zeta_s^{(m)} \in \mathtt{supp}(\nu^{(m)}),\ m = 1,\dots,M \right\}. \tag{2.6}$$

b) There exists a sparse solution $\bar{\mu}$ such that

$$|\mathtt{supp}(\bar{\mu})| \leq T - M + 1, \text{ where } T := \sum_{m=1}^{M} S^{(m)} \tag{2.7}$$

c) If $\nu^{(m)}$, $m = 1,\dots,M$, are supported on the same grid $K_1 \times \cdots \times K_d$-grid in $\mathbb{R}^d$, and $\alpha_m = \frac{1}{M}$ for all m, then there exits a solution μ to (2.4) supported on $(M(K_1-1)+1) \times \cdots \times (M(K_d-1)+1)$-grid, uniform in all directions.

Proof. Item a) is Proposition 1 (iii) in [6], Items b) and c) are Theorem 2 and Corollary 1 in the same paper. □

Let $R := |\Xi|$ be the number of points ξ in the finite set Ξ. It follows from item a) that any solution μ to problem (2.4) defined with discrete measures has the form

$$\mu = \sum_{r=1}^{R} p_r \delta_{\xi_r}, \quad \text{with} \quad p \in \Delta_R.$$

By letting $\mathcal{P}_\Xi(\mathbb{R}^d) := \left\{\mu \in \mathcal{P}(\mathbb{R}^d) : \texttt{supp}(\mu) \subset \Xi\right\}$, problem (2.4) can be reformulated as a finite-dimensional LP by replacing the constraint $\mu \in \mathcal{P}(\mathbb{R}^d)$ with $\mu \in \mathcal{P}_\Xi(\mathbb{R}^d)$. Indeed, by considering all the R points in Ξ, problem (2.4) boils down to

$$\begin{cases} \min\limits_{p \in \Delta_R,\, \pi \geq 0} & \sum\limits_{m=1}^{M} \alpha_m \sum\limits_{r=1}^{R} \sum\limits_{s=1}^{S^{(m)}} \|\xi_r - \zeta_s^{(m)}\|^2 \pi_{rs}^{(m)} \\ \text{s.t.} & \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}, \quad s = 1, \dots, S^{(m)},\ m = 1, \dots, M \\ & \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)} = p_r, \quad r = 1, \dots, R,\ m = 1, \dots, M. \end{cases} \tag{2.8}$$

Such LP scales exponentially in the number of measures. To see that, assume that all measures have support of same cardinality S, i.e., $S^{(m)} = S$ for all $m = 1, \dots, M$: then $R = S^M$ and the LP has $MRS + R = M(S)^{M+1} + (S)^M$ variables and $M(R + S) = M(S)^M + MS$ equality constraints. When the measures are supported on the same discrete grid in $\mathbb{R}^d$ and $\alpha_m = \frac{1}{M}$ for all m, the number of different points in Ξ reduces drastically: in this case, $S = K^d$ and $R = ((K-1)M + 1)^d$ from item c) above, which is significantly smaller than $S^M = K^{dM}$ in the previous general setting (however still colossal in real-life applications) [22, 23]. These observations shed light on how the number of measures, the sizes of their support sets, and dimension d impact the size of problem (2.8). The paper [23] investigates the complexity of computing a sparse Wasserstein barycenter, and [3] shows that a WB cannot be computed in time polynomial in the number of measures, (maximum) support size, and dimension d.

Item b) ensures that a sparse solution exists with a support size of at most $M(S-1)+1$, motivating thus the so-called *fixed-support* approaches that generally employ a block-coordinate optimization heuristic: at iteration k, a support Ξ^k of size R (say $R \leq M(S-1)+1$) is fixed and the LP (2.8) (with $\xi_r \in \Xi^k$) is solved to get an optimal plan π^k, which is in turn fixed in the optimization problem $\min_\xi \sum_{m=1}^{M} \alpha_m \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} \|\xi_r - \zeta_s^{(m)}\|^2 \pi_{rs}^{k,(m)}$ yielding a new fixed support Ξ^{k+1}. Observe that this last problem has a straightforward solution (see for instance [38, Alg. 2] and [131, § II]). Otherwise, when the *free-support* approach is taken, computing a WB amounts to solve (2.8) by considering all points $\xi \in \Xi$, thus yielding an LP of astronomical size. Hence, whether an exact (free-support) or inexact (fixed-support) approach is employed to compute (approximate) a WB, one invariably has to face a large-scale LP of the form (2.8), which fits into the structure of (2.1) by dropping the decision variable[2] p, setting

$$\Pi^{(m)} := \left\{ \pi^{(m)} \geq 0 : \sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)},\ s = 1, \dots, S^{(m)} \right\},\ m = 1, \dots, M, \tag{2.9}$$

and the linear subspace

$$\mathcal{B} := \left\{ \pi = (\pi^{(1)}, \dots, \pi^{(M)}) \;\middle|\; \begin{array}{lcll} \sum_{s=1}^{S^{(1)}} \pi_{rs}^{(1)} & = & \sum_{s=1}^{S^{(2)}} \pi_{rs}^{(2)}, & r = 1, \dots, R \\ \sum_{s=1}^{S^{(2)}} \pi_{rs}^{(2)} & = & \sum_{s=1}^{S^{(3)}} \pi_{rs}^{(3)}, & r = 1, \dots, R \\ & \vdots & & \\ \sum_{s=1}^{S^{(M-1)}} \pi_{rs}^{(M-1)} & = & \sum_{s=1}^{S^{(M)}} \pi_{rs}^{(M)}, & r = 1, \dots, R \end{array} \right\}. \tag{2.10}$$

[2] Although variable p is the one of interest, it can be removed from (2.8) and easily recovered thanks to the balanced subspace (2.10).

The polytope $\Pi^{(m)}$ is composed of transportation plans $\pi^{(m)}$ with right marginals $q^{(m)}$. The set with all left marginals is characterized by the linear subspace $\mathcal{B}$ of "balanced plans".

In light of the above observations, we focus on a decomposition technique for solving LPs of the form (2.1) to render computing a (free or fixed support) WB possible beyond moderate-scale data inputs. We mention in passing that no assumption on the costs of (2.1) is required. This fact opens the way to consider, for instance, W_ι^ι Wasserstein distances with $\iota \in [1, \infty)$.

2.3 Discrete unbalanced Wasserstein barycenters

A well-known drawback of formulation (2.4) is its limitation to measures with equal total masses, so the feasible set defining the Wasserstein distance (WD) is nonempty. To overcome this limitation, an idea is to relax the marginal constraints in (WD) to cope with "unbalanced" measures, i.e., with different masses [107]. Different manners to relax these marginal constraints yield different generalizations of the concept of Wasserstein barycenter, known in the literature by the name of *unbalanced Wasserstein barycenters* (UWBs). In this work, we propose a new formulation that uses a metric to measure the distance of a multi-plan $\pi = (\pi^{(1)}, \dots, \pi^{(M)})$ to the balanced subspace $\mathcal{B}$ defined in eq. (2.10). In the following we take such a metric as being the Euclidean distance $\texttt{dist}_{\mathcal{B}}(\pi) = \min_{\theta \in \mathcal{B}} \|\theta - \pi\|$ and define the following nonlinear optimization problem, with $\gamma > 0$ a penalty parameter, $\Pi^{(m)}$ given in (2.9), and $\mathcal{B}$ in (2.10):

$$\begin{cases} \min\limits_{\pi} & \sum\limits_{m=1}^{M} \alpha_m \sum\limits_{r=1}^{R} \sum\limits_{s=1}^{S^{(m)}} \|\xi_r - \zeta_s^{(m)}\|^2 \pi_{rs}^{(m)} + \gamma \texttt{dist}_{\mathcal{B}}(\pi) \\ \text{s.t.} & \pi^{(m)} \in \Pi^{(m)},\ m = 1, \dots, M. \end{cases} \tag{2.11}$$

This problem has always a solution because the objective function is continuous and the non-empty feasible set is compact. Note that in the balanced case, problem eq. (2.11) is a relaxation of eq. (2.1). In the unbalanced setting, any feasible point to eq. (2.11) yields $\texttt{dist}_{\mathcal{B}}(\pi) > 0$. As this distance function is strictly convex outside $\mathcal{B}$, the above problem has a unique solution.

Definition 7 (Discrete Unbalanced Wassertein Barycenter - UWB)**.** *Given a set $\{\nu^{(1)}, \dots, \nu^{(M)}\}$ of unbalanced non-negative vectors, let $\bar{\pi} \geq 0$ be the unique solution to problem (2.11), and $\tilde{\pi}$ the projection of $\bar{\pi}$ onto the balanced subspace $\mathcal{B}$, that is, $\tilde{\pi} := \texttt{Proj}_{\mathcal{B}}(\bar{\pi})$. The measure $\mu = \sum_{r=1}^{R} p_r \delta_{\xi_r}$ with $p_r := \sum_{s=1}^{S(m)} \tilde{\pi}_{rs}^{(m)}, r = 1, \dots, R$ for any $m \in \{1, \dots, M\}$, is defined as the γ-unbalanced Wasserstein barycenter of $\{\nu^{(1)}, \dots, \nu^{(M)}\}$.*

The above definition differs from the ones found in the literature, which relaxes the constraints $\sum_{r=1}^{R} \pi_{rs}^{(m)} = q_s^{(m)}$, see for instance [63, 107]. Although the above definition is not as general as the ones of the latter references, it provides meaningful results (see Section 2.6.4 below), uniqueness of the barycenter (if unbalanced), and is indeed an extension of (balanced) WB as the LP (2.8) is for (2.1) what the nonlinear problem (2.11) is for (2.2).

Proposition 1. *Suppose that $\{\nu^{(1)}, \dots, \nu^{(M)}\}$ are probability measures and let $\gamma > \|c\|$ in problem eq. (2.2). Then $\bar{\pi}$ solves (2.2) if and only if $\bar{\pi}$ solves (2.1). In particular, any UWB according to definition 7 is also a (balanced) WB.*

Proof. Being a linear function, the objective of (2.1) is Lipschitz continuous with constant $\|c\|$. Thus, the standard

theory of exact penalty methods in optimization (see for instance [17, Prop. 1.5.2]) ensures that, when $\gamma > \|c\|$, $\bar{\pi}$ solves problem eq. (2.2) if and only if $\bar{\pi}$ solves eq. (2.1). In particular, for $\Pi^{(m)}$ and $\mathcal{B}$ given in (2.9) and (2.10), respectively, $\bar{\pi} = \texttt{Proj}_{\mathcal{B}}(\bar{\pi})$ and the measure $\mu = \sum_{r=1}^{R} p_r \delta_{\xi_r}$ with $p_r = \sum_{s=1}^{S(m)} \bar{\pi}_{rs}^{(m)}$ (as in definition 7) solves (2.8). □

Another advantage of definition 7 is that the problem yielding the proposed UBW enjoys a favorable structure that can be efficiently exploited by splitting methods. Indeed, it turns out that computing a balanced or unbalanced WB can be done by the algorithm presented in Section 2.5.3. In the next section, we show that the computational burden to solve either the LP eq. (2.1) or the nonlinear problem eq. (2.2) by the Douglas-Rachford splitting method is the same.

2.4 Problem reformulation and the Douglas-Rachford algorithm

We have recalled that computing a free or fixed-support (balanced) WB requires solving one or more LPs of the form (2.1), with $\Pi^{(m)}$ and $\mathcal{B}$ given in (2.9) and (2.10), respectively. Furthermore, according to our new Definition 7, computing a UWB requires solving one (or more) nonlinear problems of the form (2.2). In this section, we focus on problems eq. (2.1) and eq. (2.2) and reformulate them in a suitable way so that the Douglas-Rachford splitting operator method can be easily deployed to compute a discrete barycenter in the balanced and unbalanced settings. To this end, let us consider the indicator function $\mathbf{i}_C$ of a convex set C (that is $\mathbf{i}_C(x) = 0$ if $x \in C$ and $\mathbf{i}_C(x) = \infty$ otherwise) to define the convex functions

$$f^{(m)}(\pi^{(m)}) := \sum_{r=1}^{R} \sum_{s=1}^{S^{(m)}} c_{rs}^{(m)} \pi_{rs}^{(m)} + \mathbf{i}_{\Pi^{(m)}}(\pi^{(m)}), \quad m = 1, \dots, M, \tag{2.12}$$

and recast problems eq. (2.1) and eq. (2.2) in the following more general setting

$$\min_{\pi} \; f(\pi) + g(\pi)\,, \;\text{with}: \tag{2.13a}$$

$$f(\pi) := \sum_{m=1}^{M} f^{(m)}(\pi^{(m)}) \quad \text{and} \quad g(\pi) := \begin{cases} \mathbf{i}_{\mathcal{B}}(\pi) & \text{if balanced} \\ \gamma\, \texttt{dist}_{\mathcal{B}}(\pi) & \text{if unbalanced.} \end{cases} \tag{2.13b}$$

Since f is polyhedral, $\mathcal{D}\text{om}(f) \cap \texttt{ri}(\mathcal{D}\text{om}(g)) \neq \emptyset$, and (2.13) is solvable[3], it follows from [8, Thm 27.2] that computing one of its solutions is equivalent to

$$\text{find} \quad \pi \quad \text{such that} \quad 0 \in \partial f(\pi) + \partial g(\pi). \tag{2.14}$$

Recall that the subdifferential of a proper convex lower semicontinuous functions is a maximal monotone operator [8, Thm 20.40]. Thus, the above generalized equation is nothing but the problem of finding a zero of the sum of two maximal monotone operators, a well-understood problem for which several methods exist (see, for instance, Chapters 25 and 27 of the textbook [8]). Among the existing algorithms, the Douglas-Rachford operator splitting method [50] (see also [8, § 25.2 and § 27.2]) is the most popular one. When applied to problem eq. (2.14), the

[3] $\texttt{ri}$ denotes the relative interior of a set.

DR algorithm asymptotically computes a solution by repeating the following steps, with $k = 0, 1, \ldots$, given initial point $\theta^0 = (\theta^{(1),0}, \ldots, \theta^{(M),0})$ and prox-parameter $\rho > 0$:

$$\left\{\begin{array}{lcl} \pi^{k+1} & = & \arg\min\limits_{\pi} \; g(\pi) + \dfrac{\rho}{2}\|\pi - \theta^k\|^2 \\ \hat{\pi}^{k+1} & = & \arg\min\limits_{\pi} \; f(\pi) + \dfrac{\rho}{2}\|\pi - (2\pi^{k+1} - \theta^k)\|^2 \\ \theta^{k+1} & = & \theta^k + \hat{\pi}^{k+1} - \pi^{k+1}. \end{array}\right. \tag{2.15}$$

By noting that f and g in eq. (2.13b) are proper convex lower semicontinuous functions and problem eq. (2.13) is solvable (so is (2.14) [8, Thm 27.2(ii)]), the following is a direct consequence of Theorem 25.6 and Corollary 27.4 of [8].

Theorem 2. *The sequence $\{\theta^k\}$ produced by the DR algorithm eq. (2.15) converges to a point $\bar{\theta}$, and the following holds: $\bar{\pi} := \arg\min_\pi \; g(\pi) + \frac{\rho}{2}\|\pi - \bar{\theta}\|^2$ solves eq. (2.13), and $\{\pi^k\}$ and $\{\hat{\pi}^k\}$ converge to $\bar{\pi}$.*

The DR algorithm is attractive when the two first steps in eq. (2.15) are convenient to execute, which is the case in our settings. As we will shortly see, the iterate π^{k+1} above has an explicit formula in both balanced and unbalanced cases, and computing $\hat{\pi}^{k+1}$ amounts to executing a series of independent projections onto the simplex. This task can be accomplished exactly and efficiently by specialized algorithms.

Since f in eq. (2.13b) has a separable structure, the computation of $\hat{\pi}^{k+1}$ in eq. (2.15) breaks down to a series of smaller and simpler subproblems as just mentioned. Hence, we may exploit such a structure by combining recent developments in DR's literature to produce the following randomized version of the DR algorithm eq. (2.15), with α the vector of weights in eq. (2.4):

$$\left\{\begin{array}{lcl} \pi^{k+1} & = & \arg\min\limits_{\pi} \; g(\pi) + \dfrac{\rho}{2}\|\pi - \theta^k\|^2 \\ & & \text{Draw randomly } m \in \{1, 2, \ldots, M\} \text{ with probability } \alpha_m > 0 \\ \hat{\pi}^{(m),k+1} & = & \arg\min\limits_{\pi^{(m)}} \; f^{(m)}(\pi^{(m)}) + \dfrac{\rho}{2}\|\pi^{(m)} - (2\pi^{(m),k+1} - \theta^{(m),k})\|^2 \\ \theta^{(m'),k+1} & = & \left\{\begin{array}{ll} \theta^{(m),k} + \hat{\pi}^{(m),k+1} - \pi^{(m),k+1} & \text{if } m' = m \\ \theta^{(m'),k} & \text{if } m' \neq m. \end{array}\right. \end{array}\right. \tag{2.16}$$

The randomized DR algorithm eq. (2.16) aims at reducing the computational burden and accelerating the optimization process. Such goals can be attained in some situations, depending on the underlying problem and available computational resources. The particular choice of $\alpha_m > 0$ as the probability of picking up the m^{th} subproblem is not necessary for convergence: the only requirement is that every subproblem is picked up with a fixed and positive probability. The intuition behind our choice is that measures that play a more significant role in eq. (2.4) (i.e., higher α_m) should have more chance to be picked by the randomized DR algorithm. Furthermore, the presentation above where only one measure (subproblem) in eq. (2.16) is drawn is made for the sake of simplicity. One can perfectly split the set of measures into $nb < M$ bundles, each containing a subset of measures, and select randomly bundles instead of individual measures. Such an approach proves advantageous in a parallel computing environment with nb available machines/processors (see Figure 2.3 in the numerical section). The almost surely (i.e., with probability one) convergence of the randomized DR algorithm depicted in eq. (2.16) can be summarized as follows [68, Thm 2].

Theorem 3. *The sequence $\{\pi^k\}$ produced by the randomized DR algorithm eq. (2.16) converges almost surely to a random variable $\bar{\pi}$ taking values in the solution set of problem eq. (2.13).*

This result is a special case of a thorough analysis given in [34] (see, in particular, Remark 3.5 and Section 5 in that paper.) We note that the practical performance of the randomized scheme (2.16) depends on computational resources and is thus not always effective (see Figure 2.3 below). The deterministic and asynchronous decomposition methods in [33] provide significantly more flexibility in selecting the measure $\nu^{(m)}$ (or even part of it) activated at every iteration and thus may perform better than the randomized scheme above. As these methods do not follow the general lines of the DR algorithm, we leave the specialization of such approaches to the WB problem for future research.

In the next section, we further exploit the structure of functions f and g in eq. (2.13) and rearrange terms in the schemes eq. (2.15) and eq. (2.16) to provide an easy-to-implement and memory-efficient algorithm for computing balanced and unbalanced WBs.

2.5 The Method of Averaged Marginals (MAM)

Both deterministic and randomized DR algorithms above require evaluating the proximal mapping of the function g given in eq. (2.13b). In the balanced WB setting, g is the indicator function of $\mathcal{B}$ given in eq. (2.10), and thus π^{k+1} in (2.15) is the projection of θ^k onto $\mathcal{B}$: $\pi^{k+1} = \texttt{Proj}_{\mathcal{B}}(\theta^k)$. On the other hand, in the unbalanced WB case, $g(\cdot)$ is the penalized distance function $\gamma\,\texttt{dist}_{\mathcal{B}}(\cdot)$. Computing π^{k+1} then amounts to evaluating the proximal mapping of the distance function: $\min_\pi\ \texttt{dist}_{\mathcal{B}}(\pi) + \frac{\rho}{2\gamma}\|\pi - \theta^k\|^2$. The unique solution to this problem is given by [8, Example 24.28]

$$\pi^{k+1} = \begin{cases} \texttt{Proj}_{\mathcal{B}}(\theta^k) & \text{if } \ \rho\,\texttt{dist}_{\mathcal{B}}(\theta^k) \le \gamma \\ \theta^k + \frac{\gamma}{\rho\,\texttt{dist}_{\mathcal{B}}(\theta^k)}(\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k) & \text{otherwise.} \end{cases} \tag{2.17}$$

Hence, computing π^{k+1} in both balanced and unbalanced settings boils down to projecting onto the balanced subspace. This fact allows us to provide a unified algorithm for WB and UWB problems.

2.5.1 Projecting onto the subspace of balanced plans

In what follows we exploit the particular geometry of $\mathcal{B}$ to provide an explicit formula for projecting onto this linear subspace.

Proposition 2. *With the notation of Section 2.2, let $\theta \in \mathbb{R}^{R\times T}$,*

$$a_m := \frac{\frac{1}{S^{(m)}}}{\sum_{j=1}^{M}\frac{1}{S^{(j)}}}, \quad p^{(m)} := \left(\sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m)}\right)_{1\le r\le R}, \quad \textit{and} \quad p := \sum_{m=1}^{M} a_m p^{(m)}. \tag{2.18a}$$

The projection $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$ has the explicit form:

$$\pi_{rs}^{(m)} := \theta_{rs}^{(m)} + \frac{(p_r - p_r^{(m)})}{S^{(m)}}, \quad s = 1,\dots,S^{(m)},\ r = 1,\dots,R,\ m = 1,\dots,M. \tag{2.18b}$$

Proof. First, observe that $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$ solves the QP problem

$$\begin{cases} \min\limits_{y^{(1)},\dots,y^{(M)}} & \frac{1}{2}\sum\limits_{m=1}^{M} \|y^{(m)} - \theta^{(m),k}\|^2 \\ \text{s.t} & \sum_{s=1}^{S^{(m)}} y_{rs}^{(m)} = \sum_{s=1}^{S^{(m+1)}} y_{rs}^{(m+1)}, \quad r = 1,\dots,R,\ m = 1,\dots,M-1, \end{cases} \tag{2.19}$$

which is only coupled by the "columns" of π: there is no constraint linking $\pi_{rs}^{(m)}$ with $\pi_{r's}^{(m')}$ for $r \neq r'$ and m and m' arbitrary. Therefore, we can decompose it by rows: for $r = 1,\dots,R$, the r^{th}-row $(\pi_{r1}^{(1)},\dots,\pi_{rS^{(1)}}^{(1)},\dots,\pi_{r1}^{(M)},\dots,\pi_{rS^{(M)}}^{(M)})$ of π is the unique solution to the problem

$$\begin{cases} \min\limits_{w} & \frac{1}{2}\sum\limits_{m=1}^{M}\sum\limits_{s=1}^{S^{(m)}} \left(w_s^{(m)} - \theta_{rs}^{(m)}\right)^2 \\ \text{s.t} & \sum_{s=1}^{S^{(m)}} w_s^{(m)} = \sum_{s=1}^{S^{(m+1)}} w_s^{(m+1)},\ m = 1,\dots,M-1. \end{cases} \tag{2.20}$$

The Lagrangian function of this problem is, for a dual variable u, given by

$$L_r(w,u) = \frac{1}{2}\sum_{m=1}^{M}\sum_{s=1}^{S^{(m)}} \left(w_s^{(m)} - \theta_{rs}^{(m)}\right)^2 + \sum_{m=1}^{M-1} u^{(m)}\Big(\sum_{s=1}^{S^{(m)}} w_s^{(m)} - \sum_{s=1}^{S^{(m+1)}} w_s^{(m+1)}\Big). \tag{2.21}$$

A primal-dual (w,u) solution to problem eq. (2.20) must satisfy the Lagrange system, in particular $\nabla_w L_r(w,u) = 0$ with w the r^{th} row of $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$, that is,

$$\begin{cases} \pi_{rs}^{(1)} - \theta_{rs}^{(1)} & + & u^{(1)} & = 0 & s = 1,\dots,S^{(1)} \\ \pi_{rs}^{(2)} - \theta_{rs}^{(2)} & + & u^{(2)} - u^{(1)} & = 0 & s = 1,\dots,S^{(2)} \\ & \vdots & & & \\ \pi_{rs}^{(M-1)} - \theta_{rs}^{(M-1)} & + & u^{(M-1)} - u^{(M-2)} & = 0 & s = 1,\dots,S^{(M-1)} \\ \pi_{rs}^{(M)} - \theta_{rs}^{(M)} & - & u^{(M-1)} & = 0 & s = 1,\dots,S^{(M)}. \end{cases} \tag{2.22}$$

Let us denote $p_r = \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)}$ (no matter $m \in \{1,\dots,M\}$ because $\pi \in \mathcal{B}$), $p_r^{(m)} = \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m)}$ (the r^{th} component of $p^{(m)}$ as defined in eq. (2.18a)), and sum over s the first row of system eq. (2.22) to get

$$p_r - p_r^{(1)} + u^{(1)}S^{(1)} = 0 \quad \Rightarrow \quad u^{(1)} = \frac{p_r^{(1)} - p_r}{S^{(1)}}, \tag{2.23}$$

Now, by summing the second row in eq. (2.22) over s we get

$$p_r - p_r^{(2)} + u^{(2)}S^{(2)} - u^{(1)}S^{(2)} = 0 \quad \Rightarrow \quad u^{(2)} = u^{(1)} + \frac{p_r^{(2)} - p_r}{S^{(2)}}. \tag{2.24}$$

By proceeding in this way and setting $u^{(0)} := 0$ we obtain

$$u^{(m)} = u^{(m-1)} + \frac{p_r^{(m)} - p_r}{S^{(m)}}, \quad m = 1,\dots,M-1. \tag{2.25a}$$

Furthermore, for $M-1$ we get the alternative formula $u^{(M-1)} = -\frac{p_r^{(M)} - p_r}{S^{(M)}}$. Given these dual values, we can use eq. (2.22) to conclude that the r^{th} row of $\pi = \texttt{Proj}_{\mathcal{B}}(\theta)$ is given as in eq. (3.7). What remains is to show that $p_r = \sum_{s=1}^{S^{(m)}} \pi_{rs}^{(m)}$, as defined above, is alternatively given by eq. (2.18a). To this end, observe that $u^{(M-1)} = u^{(M-1)} - u^{(0)} = \sum_{m=1}^{M-1}(u^{(m)} - u^{(m-1)})$, so:

$$u^{(M-1)} = \sum_{m=1}^{M-1}\left(\frac{p_r^{(m)} - p_r}{S^{(m)}}\right) = \sum_{m=1}^{M-1}\frac{p_r^{(m)}}{S^{(m)}} - p_r\sum_{m=1}^{M-1}\frac{1}{S^{(m)}}. \tag{2.26}$$

Recall that $u^{(M-1)} = \frac{p_r - p_r^{(M)}}{S^{(M)}}$, i.e., $p_r = p_r^{(M)} + u^{(M-1)}S^{(M)}$. Replacing $u^{(M-1)}$ with the expression eq. (2.26) yields

$$p_r = S^{(M)}\Big[\frac{p_r^{(M)}}{S^{(M)}} + u^{(M-1)}\Big] = S^{(M)}\Big[\frac{p_r^{(M)}}{S^{(M)}} + \sum_{m=1}^{M-1}\frac{p_r^{(m)}}{S^{(m)}} - p_r\sum_{m=1}^{M-1}\frac{1}{S^{(m)}}\Big], \tag{2.27}$$

which implies $p_r\sum_{m=1}^{M}\frac{1}{S^{(m)}} = \sum_{m=1}^{M}\left(\frac{p_r^{(m)}}{S^{(m)}}\right)$. Hence, p is as given in eq. (2.18a), and the proof is complete. □

Note that projection can be computed in parallel over the rows, and the average p of the marginals $p^{(m)}$ is the gathering step between parallel processors.

2.5.2 Evaluating the proximal mapping of transportation costs

In this subsection we turn our attention to the DR algorithm's second step, which requires solving a convex optimization problem of the form: $\min_\pi\ f(\pi) + \frac{\rho}{2}\|\pi - y\|^2$ (see eq. (2.15)). Given the additive structure of f in eq. (2.13b), the above problem can be decomposed into M smaller ones

$$\min_{\pi^{(m)}}\ f^{(m)}(\pi^{(m)}) + \frac{\rho}{2}\|\pi^{(m)} - y^{(m)}\|^2, \quad m = 1,\dots,M. \tag{2.28}$$

Then looking closely at every subproblem above, we can see that we can decompose it even more: the columns of the transportation plan $\pi^{(m)}$ are independent in the minimization. Besides, as the following result shows, every column optimization is simply the projection of an R-dimensional vector onto the simplex Δ_R.

Proposition 3. *The minimization $\hat{\pi} := \min_\pi\ f(\pi) + \frac{\rho}{2}\|\pi - y\|^2$ can be performed exactly, in parallel along the columns of each transport plan $y^{(m)}$, as follows: for all $m \in \{1,\dots,M\}$,*

$$\begin{pmatrix}\hat{\pi}_{1s}^{(m)}\\ \vdots \\ \hat{\pi}_{Rs}^{(m)}\end{pmatrix} = \texttt{Proj}_{\Delta_R(q_s^{(m)})}\begin{pmatrix}y_{1s} - \frac{1}{\rho}c_{1s}^{(m)}\\ \vdots \\ y_{Rs} - \frac{1}{\rho}c_{Rs}^{(m)}\end{pmatrix}, \quad s = 1,\dots,S^{(m)}. \tag{2.29}$$

Proof. It has already been argued that evaluating the proximal mapping (2.28) (split into M smaller subproblems) boils down to solving a quadratic program due to the definition of $f^{(m)}$ in (2.12):

$$\min_{\pi^{(m)}\geq 0} \sum_{r=1}^{R}\sum_{s=1}^{S^{(m)}} \left[c_{rs}^{(m)}\pi_{rs}^{(m)} + \frac{\rho}{2}\left|\pi_{rs}^{(m)} - y_{rs}^{(m)})\right|^2\right] \quad \text{s.t.} \quad \sum_{r=1}^{R}\pi_{rs}^{(m)} = q_s^{(m)},\ s=1,\dots,S^{(m)}. \tag{2.30}$$

By taking a close look at the above problem, we can see that the objective function is decomposable, and the constraints couple only the "rows" of $\pi^{(m)}$. Therefore, we can go further and decompose the above problem per columns: for $s=1,\dots,S^{(m)}$, the s^{th}-column of $\hat{\pi}^{(m)}$ is the unique solution to the R-dimensional problem

$$\min_{w\geq 0} \sum_{r=1}^{R}\left[c_{rs}^{(m)}w_r + \frac{\rho}{2}(w_r - y_{rs}^{(m)})^2\right] \quad \text{s.t.} \quad \sum_{r=1}^{R} w_r = q_s^{(m)}, \tag{2.31}$$

which is nothing but eq. (2.29). Such projection can be performed exactly [35]. □

Remark 1. *If $\tau = 0$, then $\Delta_R(\tau) = \{0\}$ and the projection onto this set is trivial. Otherwise, $\tau > 0$ and computing $\texttt{Proj}_{\Delta_R(\tau)}(w)$ amounts to projecting onto the $R+1$ simplex Δ_R: $\texttt{Proj}_{\Delta_R(\tau)}(w) = \tau\,\texttt{Proj}_{\Delta_R}(w/\tau)$. The latter task can be performed* exactly *by using efficient methods [35]. Hence, evaluating the proximal mapping in proposition 3 decomposes into T independent projections onto Δ_R.*

2.5.3 The method of averaged marginals

Our approach is presented in Algorithm 1, which gathers the three main steps from the DR algorithm and integrates a choice of γ for a simple switch between the balanced and unbalanced cases.

MAM's interpretation At every iteration, the barycenter approximation p^k is a weighted average of the M marginals $p^{(m)}$ of the plans $\theta^{(m),k}$, $m=1,\dots,M$. As we will shortly see, the whole sequence $\{p^k\}$ converges (almost surely or deterministically) to a barycenter upon specific assumptions on the choice of the index set at line 6 of Algorithm 1.

Initialization The choices for $\theta^0 \in \mathbb{R}^{R\times T}$ and $\rho > 0$ are arbitrary ones. The prox-parameter $\rho > 0$ is borrowed from the DR algorithm, which is known to have an impact on the practical convergence speed. Therefore, ρ should be tuned for the set of distributions at stake. Some heuristics for tuning this parameter exist for other methods derived from the DR algorithms [126, 129] and can be adapted to the setting of Algorithm 1.

Stopping criteria A possible stopping test is $\|\theta^{k+1} - \theta^k\|_\infty \leq \texttt{Tol}$, where $\texttt{Tol} > 0$ is a given tolerance. Alternatively, we may stop the algorithm when $\|p^{k+1} - p^k\| \leq \texttt{Tol}$. or $\text{dist}_B(\hat{\pi}^k) \leq \texttt{Tol}$. These latter tests should be understood as heuristic criteria.

Algorithm 1 Method of Averaged Marginals - MAM

▷ Step 0: input

1: Given $\rho > 0$, the cost matrix and initial point $c, \theta^0 \in \mathbb{R}^{R\times T}$, and $a \in \Delta_M$ as in eq. (2.18a), set $k \leftarrow 0$ and $p_r^{(m)} \leftarrow \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m),0}$, $r = 1, \dots, R$, $m = 1, \dots, M$

2: Set $\gamma \leftarrow \infty$ if $q^{(m)} \in \mathbb{R}_+^{S^{(m)}}$, $m = 1, \dots, M$, are balanced; otherwise, choose $\gamma \in [0, \infty)$

3: **while** not converged **do**

▷ Step 1: average the marginals

4: Compute $p^k \leftarrow \sum_{m=1}^M a_m p^{(m)}$

5: Set $t^k = 1$ if $\rho \sqrt{\sum_{m=1}^M \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}} \leq \gamma$; otherwise, $t^k \leftarrow \gamma / \left(\rho \sqrt{\sum_{m=1}^M \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}} \right)$

6: Choose an index set $\emptyset \neq \mathcal{M}^k \subseteq \{1, \dots, M\}$

7: **for** $m \in \mathcal{M}^k$ **do**

▷ Step 2: update the m^{th} plan

8: **for** $s = 1, \dots, S^{(m)}$ **do**

9: Define $w_r \leftarrow \theta_{rs}^{(m),k} + 2\, t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}} - \frac{1}{\rho} c_{rs}^{(m)}$, $r = 1, \dots, R$

10: Compute $(\hat{\pi}_{1s}^{(m)}, \dots, \hat{\pi}_{Rs}^{(m)}) \leftarrow \mathtt{Proj}_{\Delta_R(q_s^{(m)})}(w)$

11: Update $\theta_{rs}^{(m),k+1} \leftarrow \hat{\pi}_{rs}^{(m)} - t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}}$, $r = 1, \dots, R$

12: **end for**

▷ Step 3: update the m^{th} marginal

13: Update $p_r^{(m)} \leftarrow \sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m),k+1}$, $r = 1, \dots, R$

14: **end for**

15: **end while**

16: Return $\bar{p} \leftarrow p^k$

Deterministic and random variants of MAM The most computationally expensive step of MAM is Step 2, which requires a series of independent projections onto the $R+1$ simplex (see remark 1). Our approach underlines that this step can be conducted in parallel over s or, if preferable, over the measures m. As a result, it is a natural idea to derive a randomized variant of the algorithm. This is the reason for having the possibility of choosing an index set $\mathcal{M}^k \subsetneq \{1, \dots, M\}$ at line 6 of algorithm 1. For example, we may employ an economical rule and choose $\mathcal{M}^k = \{m\}$ randomly (with a fixed and positive probability, e.g. α_m) at every iteration, or the costly one $\mathcal{M}^k = \{1, \dots, M\}$ for all k. The latter yields the deterministic method of averaged marginals, while the former gives rise to a randomized variant of MAM. Depending on the computational resources, intermediate choices between these two extremes can perform better in practice.

Remark 2. *Suppose that $1 < \mathtt{nb} < M$ processors are available. We may then create a partition $A_1, \dots, A_{\mathtt{nb}}$ of the set $\{1, \dots, M\}$ $(= \cup_{i=1}^{\mathtt{nb}} A_i)$ and define weights $\beta_i := \sum_{m\in A_i} \alpha_m > 0$. Then, at every iteration k, we may draw with probability β_i the subset A_i of measures and set $\mathcal{M}^k = A_i$.*

This randomized variant would enable the algorithm to compute more iterations per time unit but with less precision per iteration (since not all the marginals $p^{(m)}$ are updated). Such a randomized variant of MAM is benchmarked

against its deterministic counterpart in Section 2.6.2.3, where we demonstrate empirically that with certain configurations (depending on the number M of probability distributions and the number of processors) this randomized algorithm can be effective. We highlight that other choices for $\mathcal{M}^k$ rather than randomized ones or the deterministic rule $\mathcal{M}^k = \{1, \dots, M\}$ should be understood as heuristics. Within such a framework, one may choose $\mathcal{M}^k \subsetneq \{1, \dots, M\}$ deterministically, for instance cyclically or yet by the discrepancy of the marginal $p^{(m)}$ with respect to the average p^k.

Storage complexity Note that the operation at line 8 is trivial if $q_s^{(m)} = 0$. This motivates us to remove all the zero components of $q^{(m)}$ from the problem's data, and consequently, all the columns s of the cost matrix $c^{(m)}$ and variables $\theta, \hat{\pi}$ corresponding to $q_s^{(m)} = 0$, $m = 1, \dots, M$. In some applications (e.g. general sparse problems), this strategy significantly reduces the WB problem and thus memory allocation, since the non-taken columns are both not stored and not treated in the *for loops*. This remark raises the question of how sparse data impacts the practical performance of MAM. Section 2.6.1 conducts an empirical analysis on this matter.

In nominal use, the algorithm needs to store the decision variables $\theta^{(m)} \in \mathbb{R}^{R \times S^{(m)}}$ for all $m = 1, \dots, M$ (transport plans for every measure), along with M distance matrices $c \in \mathbb{R}^{R \times S^{(m)}}$, one barycenter approximation $p^k \in \mathbb{R}^R$, M approximated marginals $p^{(m)} \in \mathbb{R}^R$ and M marginals $q^{(m)} \in \mathbb{R}^{S(m)}$. Note that in practical terms, the auxiliary variables w and $\hat{\pi}$ in algorithm 1 can be easily removed from the algorithm's implementation by merging lines 9-11 into a single one. Hence, by letting $T = \sum_{m=1}^{M} S^{(m)}$, the method's memory allocation is $2RT + T + M(R+1)$ floating-points. This number can be reduced if the measures share the same cost matrix, i.e., $c^{(m)} = c^{(m')}$ for all $m, m' = 1, \dots, M$. In this case, $S^{(m)} = S$ for all m, $T = M\,S$ and the method's memory allocation drops to $RT + RS + T + M(R+1)$ floating-points. In the light of the previous remark this memory complexity should be treated as an upper bound: the sparser the data the less memory will be needed.

Computation complexity Step 2 of the algorithm involves two main components: projection onto $\mathcal{B}$, which comprises straightforward operations detailed in Section 2.5.1, and projection onto Π, which relies on leveraging the simplex projection technique discussed in Section 2.5.2. For each probability measure, the projection onto $\mathcal{B}$ requires $3RS^{(m)}$ computation operations, where R is the barycenter support size and $S^{(m)}$ denotes the support size of the probability measure m, which undergoes iteration over its columns. On the other hand, the simplex projection (line 8) is computationally more intensive. This is due to the adoption of a state-of-the-art algorithm proposed by Condat [35], which operates in $\mathrm{O}(R\log(R))$. Therefore, the complexity of $S^{(m)}$ times line 8 amounts to $\mathrm{O}(S^{(m)}R\log(R))$. Deriving the precise number of computational operations is challenging due to the use of a sorting algorithm in [35], the complexity of which depends on the characteristics of the input data, hence the asymptotic complexity estimation. Note that Step 2 must be executed for the M probability measures, but these operations can be performed in parallel (multiprocessing).

Balanced and unbalanced settings As already mentioned, our approach can handle both balanced and unbalanced WB problems. All that is necessary is to choose a finite (positive) value for the parameter γ in the unbalanced case. Such a parameter is only used to define $t^k \in (0, 1]$ at every iteration. Indeed, algorithm 1 defines $t^k = 1$ for all iterations if the WB problem is balanced (because $\gamma = \infty$ in this case)[4], and $t^k = \gamma / \left(\rho \sqrt{\sum_{m=1}^{M} \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}} \right)$

[4]Observe that line 5 can be entirely disregarded in this case, by setting $t^k = t = 1$ fixed at initialization.

otherwise. This rule for setting up t^k is a mere artifice to model eq. (2.17). Indeed, $\texttt{dist}_{\mathcal{B}}(\theta^k) = \|\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k\|$ reduces to $\sqrt{\sum_{m=1}^{M} \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}}$ thanks to proposition 2.

Convergence analysis The convergence analysis of algorithm 1 can be summarized as follows.

Theorem 4 (MAM's convergence analysis)**.** *a) (Deterministic MAM.) Consider algorithm 1 with the choice $\mathcal{M}^k = \{1, \dots, m\}$ for all k. Then the sequence of points $\{p^k\}$ generated by the algorithm converges to a point $\bar{p}$. If the measures are balanced, then $\bar{p}$ is a balanced WB; otherwise, $\bar{p}$ is a γ-unbalanced WB.*

b) (Randomized MAM.) Consider algorithm 1 with the choice $\mathcal{M}^k \subset \{1, \dots, m\}$ as in remark 2. Then the sequence of points $\{p^k\}$ generated by the algorithm converges almost surely to a point $\bar{p}$. If the measures are balanced, then $\bar{p}$ is almost surely a balanced WB; otherwise, $\bar{p}$ is almost surely a γ-unbalanced WB.

Proof. It suffices to show that algorithm 1 is an implementation of the (randomized) DR algorithm and invoke Theorem 2 for item a) and Theorem 3 for item b). To this end, we first rely on proposition 2 to get that the projection of θ^k onto the balanced subspace $\mathcal{B}$ is given by $\theta_{rs}^{(m),k} + \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}$, $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, $m = 1, \dots, M$, where p^k is computed at Step 1 of the algorithm, and the marginals $p^{(m)}$ of θ^k are computed at Step 0 if $k = 0$ or at Step 3 otherwise. Therefore, $\texttt{dist}_{\mathcal{B}}(\theta^k) = \|\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k\| = \sqrt{\sum_{m=1}^{M} \frac{\|p^k - p^{(m)}\|^2}{S^{(m)}}}$. Now, given the rule for updating t^k in algorithm 1 we can define the auxiliary variable π^{k+1} as $\pi^{k+1} = \theta^k + t^k(\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k)$, or alternatively,

$$\pi_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}, \quad s = 1, \dots, S^{(m)},\ r = 1, \dots, R,\ m = 1, \dots, M. \tag{2.32}$$

In the balanced case, $t^k = 1$ for all k (because $\gamma = \infty$) and thus π^{k+1} is as in eq. (3.7). Otherwise, π^{k+1} is as in eq. (2.17) (see the comments after algorithm 1). In both cases, π^{k+1} coincides with the auxiliary variable at the first step of the DR scheme eq. (2.15) (see the developments at the beginning of this section). Next, observe that to perform the second step of eq. (2.15) we need to assess $y = 2\pi^{k+1} - \theta^k$, which is thanks to the above formula for π^{k+1} given by $y_{rs}^{(m)} = \theta_{rs}^{(m),k} + 2\, t^k \frac{p_r^k - p_r^{(m)}}{S^{(m)}}$, $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, $m = 1, \dots, M$.

As a result, for the choice $\mathcal{M}^k = \{1, \dots, M\}$ for all k, Step 2 of algorithm 1 yields, thanks to proposition 3, $\hat{\pi}^{k+1}$ as at the second step of eq. (2.15). Furthermore, the updating of θ^{k+1} in the latter coincides with the rule in algorithm 1: for $s = 1, \dots, S^{(m)}$, $r = 1, \dots, R$, and $m = 1, \dots, M$,

$$\begin{aligned}\theta_{rs}^{(m),k+1} &= \theta_{rs}^{(m),k} + \hat{\pi}_{rs}^{(m),k+1} - \pi_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + \hat{\pi}_{rs}^{(m),k+1} - \left(\theta_{rs}^{(m),k} + t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}\right) \\ &= \hat{\pi}_{rs}^{(m),k+1} - t^k \frac{(p_r^k - p_r^{(m)})}{S^{(m)}}.\end{aligned}$$

Hence, for the choice $\mathcal{M}^k = \{1, \dots, M\}$ for all k, algorithm 1 is the DR Algorithm eq. (2.15) applied to the WB eq. (2.13). Theorem 2 thus ensures that the sequence $\{\pi^k\}$ as defined above converges to some $\bar{\pi}$ solving eq. (2.13). To show that $\{p^k\}$ converges to a barycenter, let us first use the property that $\mathcal{B}$ is a linear subspace to obtain the decomposition $\theta = \texttt{Proj}_{\mathcal{B}}(\theta) + \texttt{Proj}_{\mathcal{B}^\perp}(\theta)$ that allows us to rewrite the auxiliary variable π^{k+1} differently:

$$\pi^{k+1} = \theta^k + t^k(\texttt{Proj}_{\mathcal{B}}(\theta^k) - \theta^k) = \theta^k - t^k \texttt{Proj}_{\mathcal{B}^\perp}(\theta^k).$$

Let us denote $\tilde{\pi}^{k+1} := \texttt{Proj}_{\mathcal{B}}(\pi^{k+1})$. Then $\tilde{\pi}^{k+1} = \texttt{Proj}_{\mathcal{B}}(\theta^k - t^k \texttt{Proj}_{\mathcal{B}^\perp}(\theta^k)) = \texttt{Proj}_{\mathcal{B}}(\theta^k)$, and thus proposition 2 yields

$$\tilde{\pi}_{rs}^{(m),k+1} = \theta_{rs}^{(m),k} + \frac{p_r^k - p_r^{(m)}}{S^{(m)}} \quad s = 1, \dots, S^{(m)},\ r = 1, \dots, R,\ m = 1, \dots, M,$$

which in turn gives (by recalling that $\sum_{s=1}^{S^{(m)}} \theta_{rs}^{(m),k} = p_r^{(m)}$): $\sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m),k+1} = p_r^k$, $r = 1, \dots, R,\ m = 1, \dots, M$. As $\lim_{k\to\infty} \pi^k = \bar{\pi}$, $\lim_{k\to\infty} \tilde{\pi}^k = \lim_{k\to\infty} \texttt{Proj}_{\mathcal{B}}(\pi^k) = \texttt{Proj}_{\mathcal{B}}(\bar{\pi}) =: \tilde{\pi}$. Therefore, for all $r = 1, \dots, R$, $m = 1, \dots, M$, the following limits are well defined:

$$\bar{p}_r := \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m)} = \lim_{k\to\infty} \sum_{s=1}^{S^{(m)}} \tilde{\pi}_{rs}^{(m),k+1} = \lim_{k\to\infty} p_r^k. \tag{2.33}$$

We have shown that the whole sequence $\{p^k\}$ converges to $\bar{p}$. By recalling that $\bar{\pi}$ solves eq. (2.13), we conclude that in the balanced setting $\tilde{\pi} = \bar{\pi}$ and thus $\bar{p}$ is a WB. On the other hand, in the unbalanced setting, $\bar{p}$ above is a γ-unbalanced WB according to definition 7.

The proof of item b) is a verbatim copy of the above proof: the sole difference, given the assumptions on the choice of $\mathcal{M}^k$, is that we need to rely on theorem 3 (and not on theorem 2 as previously done) to conclude that $\{\pi^k\}$ converges almost surely to some $\bar{\pi}$ solving eq. (2.13). Thanks to the continuity of the orthogonal projection onto the subspace $\mathcal{B}$, the limits above yield almost surely convergence of $\{p^k\}$ to a barycenter $\bar{p}$. □

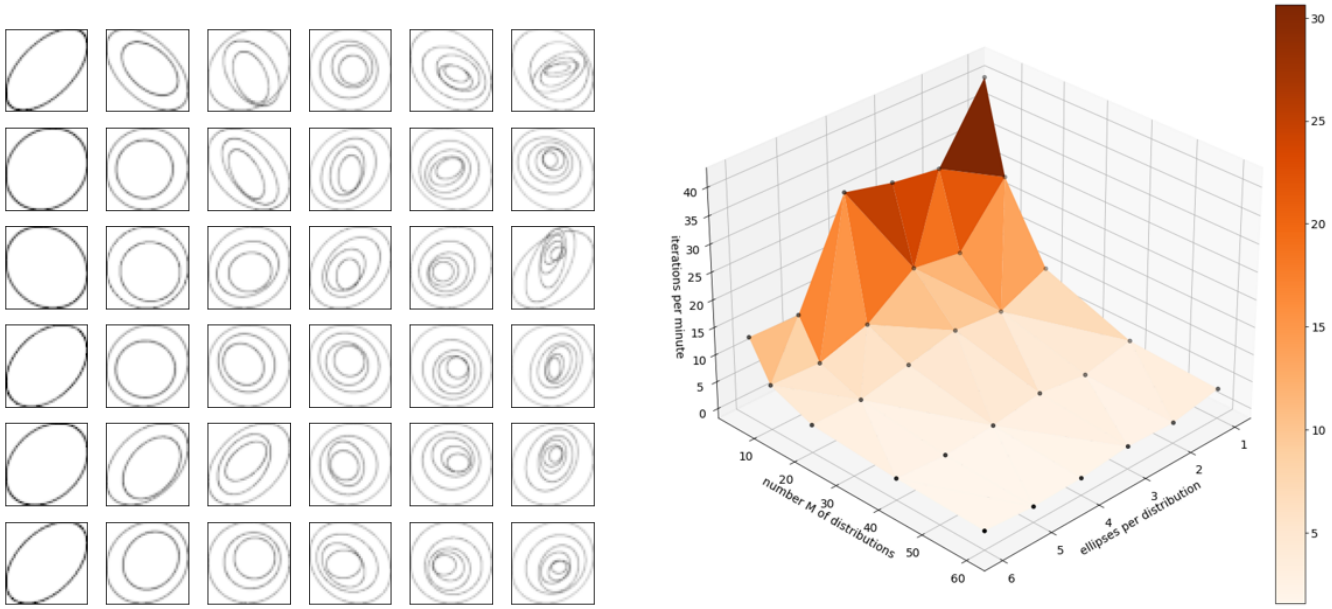
2.6 Numerical experiments

This section illustrates the MAM's practical performance on some well-known datasets. The impact of different data structures is studied before the algorithm is compared to state-of-the-art methods. This section closes with an illustrative example of MAM to compute UWBs. Numerical experiments were conducted using 20 cores (*Intel(R) Xeon(R) Gold 5120 CPU*) and *Python 3.9*. The test problems and solvers' codes are available for download in the link `https://ifpen-gitlab.appcollaboratif.fr/detocs/mam_wb`.

2.6.1 Study on data structure influence

We start by evaluating the impact of conditions that influence the storage complexity and the algorithm performance. The main conditions are the *sparsity* of the data and the *number of distributions* M. Naturally, the denser the distributions or the more distributions are treated, the greater the storage. In these configurations, the time per iteration grows because the number of projects onto the simplex increases. To assess the impact of data sparsity and the number of measures on the algorithm's performance, we consider a fixed-support approach and experiment on datasets inspired by [15, 38]. The number of nested ellipses controls the density of a dataset: as exemplified in fig. 2.1(a) and table 5.1, measures with only a single ellipse are very sparse. In contrast, a dataset with 5 nested ellipses is denser.

In this first experiment, we apply MAM with $\rho = 100$ (without proper tuning) for every dataset. Only a single processor was considered to avoid CPU communication management. Figure 2.1(b) shows that, as expected, the



(a) A sample of datasets

(b) MAM's number of iterations per second

Figure 2.1: (a) Example from the artificial nested ellipses datasets. Each column shows one dataset: the first with 1 ellipse, the second with 2 nested ellipses, up to the sixth with 6 nested ellipses. (b) Influence of density and number of distributions on MAM's iteration speed (iterations per second).

Table 2.1: Mean density with the number of nested ellipses. The density has been calculated by averaging the ratio of non-null pixels per image over 100 generated pictures for each dataset sharing the same number of nested ellipses.

Number of ellipses	1	2	3	4	5	6
Density (%)	29.0	51.4	64.3	70.9	73.5	75.0

execution time of an iteration increases with increasing density and number of measures. The number of measures influences the method's speed more than density (this phenomenon can be due to the *numpy* matrix management). This means the quantity of information in each measure does not seem to make the algorithm less efficient in terms of speed. Such a result is to be put in regard with algorithms such as B-ADMM [131] that are particularly shaped for sparse datasets but less efficient for denser ones. Section 2.6.2.4 develops this further. Additionally, it is worth noting that the proposed method can harness parallel computation, enabling the distribution of work across the M measures. This approach effectively mitigates the impact of the measure count on computational efficiency.

The growing dimensions of images have an impact on the computation time, as seen in Section 2.5.3. For example, when treating dense $K \times K$ images for a fixed support problem, the number of operations per probability density for the projection onto $\mathcal{B}$ is $O_1^{\mathcal{B}} = 3 \cdot K^2 \cdot K^2 = 3 \cdot K^4$ and onto the simplex $O_1^{\Delta} = K^2 \cdot K^2 \cdot \log(K^2) = 2K^4 \cdot \log(K)$.

- For a fixed-support problem with dense $(nK) \times (nK)$ images, $O_{nK}^{\mathcal{B}} = 3 \cdot (nK)^4 = n^4 \cdot O_1^{\mathcal{B}}$ and $O_{nK}^{\Delta} =$

$n^4K^4\log(n^2K^2) \approx n^4 \cdot O_1^{\Delta}$.

- For a fixed-support problem with dense $K \times \cdots \times K = K^d$ measures, $O_{K^d}^{\mathcal{B}} = 3 \cdot (K^d)^2 = K^{2d-4} \cdot O_1^{\mathcal{B}}$ and $O_{K^d}^{\Delta} = (K^d)^2 \log(K^d) = \frac{d}{2} K^{2d-4} \cdot O_1^{\Delta}$.
- For a free-support problem, in dimension d, with dense K^d grids, the size of the support R depends on the number M of treated measures, $R = ((K-1)M+1)^d$. Following the details of Section 2.5.3, $O_{free,K^d}^{\mathcal{B}} = 3((K-1)M+1)^d K^d \approx M^d K^{2d-4} O_1^{\mathcal{B}}$ and $O_{free,K^d}^{\Delta} = ((K-1)M+1)^d K^d \log(((K-1)M+1)^d) \approx \frac{d}{2} M^d K^{2d-4} \cdot O_1^{\Delta}$.

For instance, for a fixed-support problem with 40×40 images (see fig. 2.1), the algorithm computes the projections for one measure in an average time of 0.01 seconds. However, for the free-support problem formulation with this dataset of 6 images, it takes 6 seconds per measure. Similarly, for a fixed-support problem with $40 \times 40 \times 40$ objects (ellipsoids with similar properties as in fig. 2.1 in 3D), the projections for one measure take 16 seconds.

2.6.2 Fixed-support approach

This section focuses on the fixed-support approach: R in (2.8) is equal to K^2, the number of pixels of a $K \times K$ image.

2.6.2.1 Comparison with IBP

The Iterative Bregman Projection (IBP) [15] is a well-known algorithm for computing Wasserstein barycenters. As mentioned in the Introduction, IBP employs a regularizing function parameterized by $\lambda > 0$, which impacts precision and must kept at a moderate magnitude to avoid numerical errors (double-precision overflow). The experiment below sheds light on the differences between MAM and IBP and their advantages depending on the use. Our IBP code is inspired by the original `MATLAB` code by G. Peyré[5].

2.6.2.2 Qualitative comparison

Here, we use 100 images per digit of the MNIST database [119], where each digit has been randomly translated and rotated. Each image has 40×40 pixels and can be treated as probability distributions after normalization. Section 2.6.2.2 displays intermediate solutions for digits $3, 4, 5$ at different time steps both for MAM and IBP. For the two methods, the hyperparameters have been tuned: for instance, $\lambda = 1700$ is the greatest lambda that enables IBP to compute the barycenter of the 3's dataset without double-precision overflow error. Regarding MAM, a range of values for $\rho > 0$ have been tested for 100 seconds of execution, to identify which one provides good performance (for example, $\rho = 50$ for the dataset of 3's). Section 2.6.2.2 shows that, for each dataset, IBP gets quickly to a stable barycenter approximation. Such a point is obtained shortly after with MAM (less than 10 seconds after). However, MAM continues to move towards a sharper solution. It is clear that the more CPUs used for MAM, the better. Furthermore, while IBP is not well-suitable for CPU parallelization [15, 92, 131], MAM offers a clear advantage depending on the hardware at stake.

[5] https://github.com/gpeyre/2014-SISC-BregmanOT

2.6.2.3 Quantitative comparison

Next, we benchmark MAM, randomized MAM and IBP on a dataset with 60 images per digit of the MNIST database [119], where every digit is a normalized image 40×40 pixels. First, all three methods have their hyperparameters tuned thanks to a sensitivity study as explained in Section 2.6.2.2. Then, at every time step an approximation of the computed barycenter is stored to compute the error $\bar{W}_2^2(p^k) - \bar{W}_2^2(p_G) := \sum_{m=1}^M \frac{1}{M} W_2^2(\mu^k, \nu^{(m)}) - \sum_{m=1}^M \frac{1}{M} W_2^2(\mu_G, \nu^{(m)}))$, where μ_G is a fixed-support barycenter computed using *Gurobi* to solve the LP eq. (2.8).

For this dataset, Figure 2.3 shows that IBP is almost 10 times faster per iteration. However, IBP computes a solution to the regularized model, not to the (fixed-support) WB linear problem (2.8). Instead, MAM does converge to a solution of (2.8). So there is a threshold where the accuracy of MAM exceeds that of IBP: in our case, around 200s - for the computation with the greatest number of processors (see Figure 2.3). Such a threshold always exists depending on the computational means (hardware). This quantitative study explains what has been exemplified with the images of Section 2.6.2.2: the accuracy of IBP is bounded by the choice of λ, itself bounded by an overflow error. In contrast, the MAM hyperparameter only impacts the convergence speed. For this dataset, the WB computed by IBP is within 2% of accuracy and thus reasonably good. However, as shown in Table 1 in [131], one can choose other datasets where IBP's accuracy might be unsatisfactory.

Furthermore, Figure 2.3 exemplifies an attractive asset of randomized variants of MAM: in some configurations, randomized MAM is more efficient than (deterministic) MAM. (The curve *MAM 1-random, 1 processor* does not appear in the figure because it is above the y-axis value range due to its bad performance.) Indeed, a trade-off exists between time spent per iteration and precision gained after an iteration. For example, with 10 processors, each processor treats six measures in the deterministic MAM, but only one is treated in the randomized MAM. Therefore, the time spent per iteration is roughly six times shorter in the latter, which counterbalances the loss of accuracy per iteration. On the other hand, when using 20 processors, only three measures are treated by each processor, and the trade-off is not worth it anymore: the gain in time does not compensate for the loss in accuracy per iteration. One should adapt the use of the algorithm with care since this trade-off conclusion is only heuristic and strongly depends on the underlying dataset and hardware. A sensitivity analysis is always a good thought for choosing the most effective amount of measures handled per processor while using the randomized MAM against the deterministic MAM

2.6.2.4 Influence of the support

This section echoes Section 2.6.1 and studies the influence of the support size. To do so, two datasets have been tested for MAM and IBP. The first dataset is already used in Section 2.6.2.3: 60 pictures of 3's taken from the MNIST database [119]. The second dataset is also composed of these 60 images but each digit has been randomly translated and rotated in the same way as in section 2.6.2.2. Therefore, the union of the support of the second dataset is greater than the first one.

section 2.6.2.4 presents two graphs that have been obtained just as in Section 2.6.2.3, but displaying the evolution in percentage: $\Delta W_{\%} := \frac{\bar{W}_2^2(p^k) - \bar{W}_2^2(p_G)}{\bar{W}_2^2(p_G)} \times 100$. Once more, the hyperparameters have been fully tuned. The hyperparameter of the IBP method is smaller for the second dataset. Indeed, as stated in [131], the greater the support, the stronger the restrictions on λ, and thus, the less precise IBP.

2.6.2.5 Comparison with B-ADMM

This subsection compares MAM with the algorithm B-ADMM of [124] using the dataset and `MATLAB` implementation provided by the authors at the link `https://github.com/bobye/d2_kmeans`. We omit IBP in our analysis because it has already been shown in [124, Table I] that IBP is outperformed by B-ADMM in this dataset. As in [124, Section IV], we consider $M = 1000$ discrete measures, each with a sparse finite support set obtained by clustering pixel colors of images. The average number of support points is around 6, and the barycenter's number of fixed-support points is $R = 60$. The optimal value of (2.8) is 712.7, computed in 10.6 seconds by the Gurobi LP solver. We have coded MAM in `MATLAB` to have a fair comparison with the `MATLAB` B-ADMM algorithm provided at the above link. Since MAM and B-ADMM use different stopping tests, we have set their stopping tolerances equal to zero and let the solvers stop with a maximum number of iterations. Table 2.2 below reports CPU time in seconds and the objective values yielded by the (approximated) barycenter $\tilde{p}$ computed by both solvers: $\bar{W}_2^2(\tilde{p})$.

Table 2.2: MAM vs B-ADMM. B-ADMM code is the one provided by its designers without changing parameters (except the stopping set to zero and the maximum number of iterations). Both algorithms use the same initial point. The optimal value of the WB barycenter for this dataset is 712.7, computed by Gurobi in 10.6 seconds.

Iterations	Objective value		Seconds	
	B-ADMM	MAM	B-ADMM	MAM
100	742.8	716.7	1.1	1.1
200	725.9	714.1	2.4	2.2
500	716.5	713.3	5.6	5.4
1000	714.1	712.9	11.8	10.8
1500	713.5	712.8	18.9	16.2
2000	713.3	712.8	25.1	21.6
2500	713.2	712.8	31.0	27.1
3000	713.1	712.7	39.8	32.4

The results show that, for the considered dataset, MAM and B-ADMM are comparable regarding CPU time, with MAM providing more precise results. B-ADMM currently lacks a convergence analysis, unlike MAM.

2.6.3 Free-support approach

This section considers the *free-support* problem (see Section 2.2, theorem 1), where the measures are supported on the same discrete grid in $\mathbb{R}^2$ ($d = 2$) and $\alpha_m = \frac{1}{M}$ for all $m = 1, \dots, M$. The dataset we use is the one from [2], illustrated in Figure 2.5. In this case, $M = 10$ measures, $S = K^2 = 60^2$ and $R = ((K-1)M+1)^d = 591^2 = 349281$. The resulting LP problem is too large to be solved by standard solvers. Therefore, we employed the dedicated solver of [2], available at the link `https://github.com/eboix/high_precision_barycenters`.

Figure 2.6 presents the evolution of the points computed by MAM along 9 hours of processing. The image on the right-hand side is an exact barycenter computed by the solver of [2] after 3.5 hours. We recall that [2] handles the dual of (2.8) by employing a geometry-based separation oracle. Once the dual is solved, the method recovers a primal vertex, yielding thus a sparse WB. As a result, the right-hand side image in Figure 2.6 is sharp. Such an

exact WB is sharper than the point provided by MAM after 9 hours.

Despite the visual differences, the point provided by MAM is a Wasserstein barycenter. To see this, we compare the values of the objective function in (2.8), i.e., the Wassersein barycentric distance. The exact solution of the method in [2] has a barycentric distance of 0.2666. After 1 hour of processing, our method had a barycenter distance of 0.2702, which improved to 0.2667 after 3.5 hours, when the solver [2] halts. The slight visual difference stems from the fact that [2] finds a vertex solution to WB problem while MAM does not.

Figure 2.7 illustrates MAM's iterative process. Let $\hat{\pi}^{(m),k}$ be the m^{th} transportation plan computed by MAM at iteration k (see Line 8 of Algorithm 1). Note that $\hat{W}_2^2(p^k) := \sum_{m=1}^{M} \langle c^{(m)}, \hat{\pi}^{(m),k} \rangle$ is an approximation to $\bar{W}_2^2(p^k) := \sum_{m=1}^{M} W_2^2(\mu^k, \nu^{(m)})$, the exact function value (barycentric distance) at iteration k. Furthermore, as it can be seen from the Douglas-Rachford algorithm in eq. (2.15), the transport plans $\hat{\pi}^{(m),k}$ do not necessarily respect the constraint embodied by $\mathcal{B}$. Thus, the approximate value $\hat{W}_2^2(p^k)$ has to be seen in perspective with the distance of $\hat{\pi}^k$ to $\mathcal{B}$, i.e., $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k)$. Figure 2.7 shows the evolution of the approximate barycentric distance $\hat{W}_2^2(p^k)$, infeasibility measure $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k)$, exact barycentric distance $\bar{W}_2^2(p^k)$, and optimal value $\bar{W}_2^2(p_{exact}) = 0.2666$. We emphasize that $\bar{W}_2^2(p^k)$ is computed (by Gurobi) after terminating MAM, while $\hat{W}_2^2(p^k)$ and $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k)$ are computed along the iterative process. After 3.5 hours, MAM provides $\bar{W}_2^2(p^k) = 0.2667$, $\hat{W}_2^2(p^k) = 0.2658$ and $\texttt{dist}_{\mathcal{B}}(\hat{\pi}^k) = 1.27 \cdot 10^{-5}$.

Because of its structure, the algorithm of [2] cannot provide intermediary approximations of the barycenters, which is a disadvantage of the method over MAM. As an example, we consider $M = 10$ images 40×40 presented in Section 2.6.2.3. Although smaller, these images are much denser than the ones in Figure 2.5 and thus the WB problem is more complicated. While MAM can provide *free-support* WB approximations all along its iterative process, the solver of [2] could not provide a solution after 50 hours of processing.

2.6.4 Unbalanced Wasserstein barycenters

This section treats a particular example to illustrate the interest in using UWB. The artificial dataset is composed of 50 images with resolution 80×80. Each image is divided into four squares. The top left, bottom left, and bottom right squares are randomly filled with double nested ellipses and the top right square is always empty as exemplified in Figure 2.8. In this example, every image is normalized to depict a probability measure so that we can compare (fixed-support) WB and UWB.

With respect to eq. (2.11), one set of constraints is relaxed and the influence of the hyperparameter γ is studied. If γ is large enough (i.e. greater than $\|\texttt{vec}(c)\| \approx 1000$, see Proposition 1), the problem boils down to the standard WB problem since the example deals with probability measures: the resulting UWB is indeed a WB. When decreasing γ the transportation costs take more importance than the distance to $\mathcal{B}$ which is more and more relaxed. Therefore, as illustrated by Figure 2.9, the resulting UWB splits the image into four parts, giving visual meaning to the fixed-support barycenter.

In the same vein, Figure 2.10 provides an illustrative application of MAM for computing UWB in another dataset.

Figure 2.2: (top) For each digit 36 out of the 100 scaled, translated, and rotated images considered for each barycenter. (bottom) Barycenters after $t = 10, 50, 500, 1000, 2000$ seconds, where the left-hand-side is the IBP evolution of its barycenter approximation, the middle panel is MAM's evolution using 10 CPU and the right-hand-side is a solution computed by applying *Gurobi* to the LP eq. (2.8).

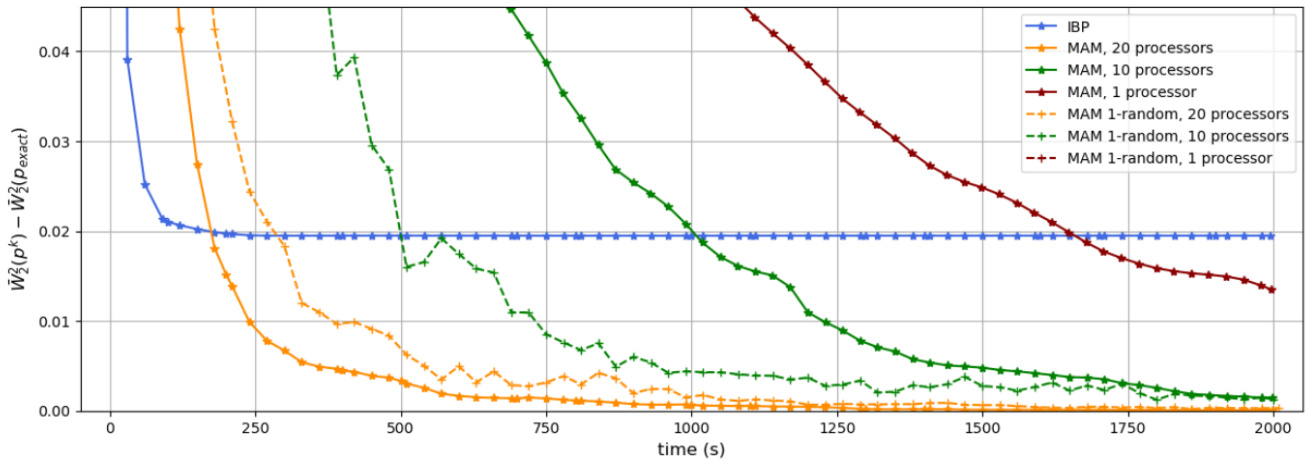


Figure 2.3: Evolution with respect to time of the difference between the Wasserstein barycenter distance of an approximation, $\bar{W}_2^2(p^k)$, and the Wasserstein barycentric distance of the exact solution $\bar{W}_2^2(p_G)$ given by the LP. The time step between two points is 30 seconds. The plot *MAM 1-random, 1 processor* is out of scope.

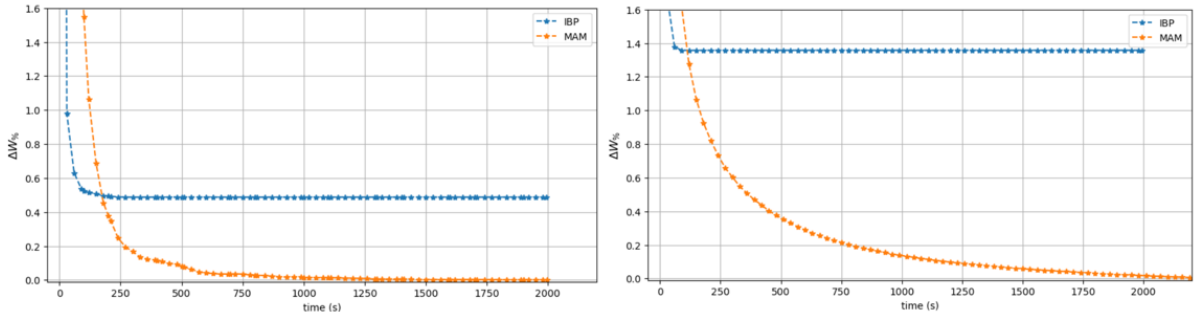


Figure 2.4: Evolution of the percentage of the distance between the exact solution of the barycenter problem and the computed solution using IBP and MAM method with 20 processors: (left) for the standard MNIST, (right) for the randomly translated and rotated MNIST.

Figure 2.5: Five 60×60 images from the nested ellipses dataset in [2].

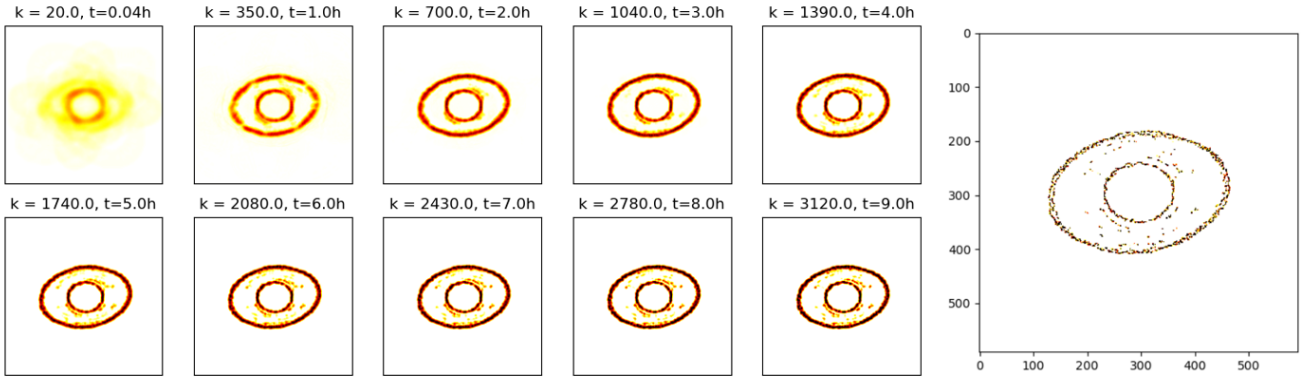


Figure 2.6: Evolution of the approximated MAM barycenter with time in regards with the exact barycenter of the Altschuler and Bois-Adsera algorithm computed in 4 hours [2].

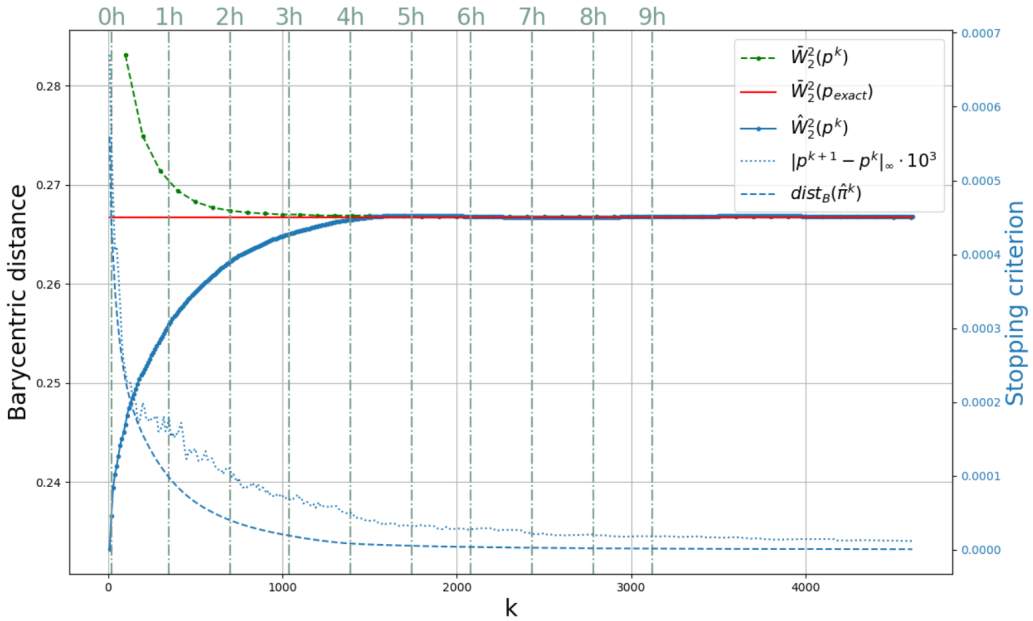


Figure 2.7: Evolution of the approximated Wasserstein barycenter distance $\hat{W}_2^2(p^k)$ with iterations (k) and time.

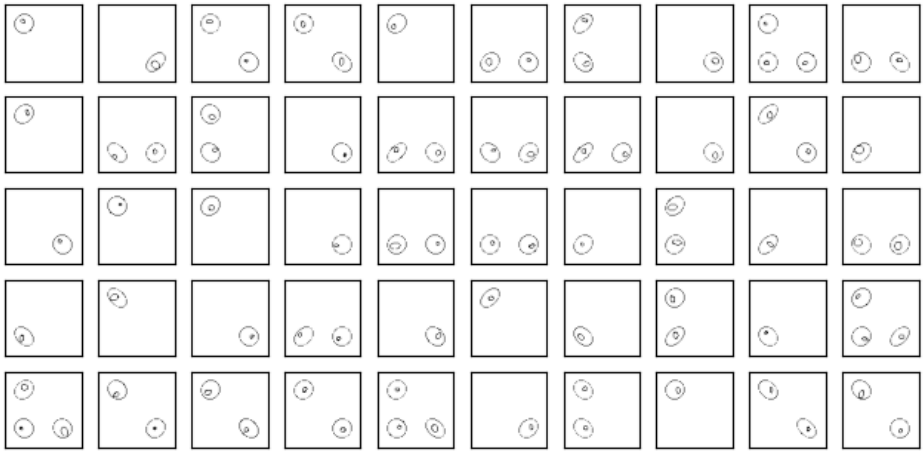


Figure 2.8: Dataset composed of 50 pictures with nested ellipses randomly positioned in the top left, bottom right, and left corners.

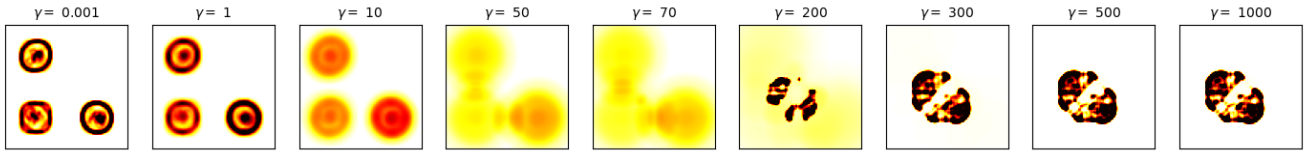


Figure 2.9: UWB computed with MAM for different values of γ.

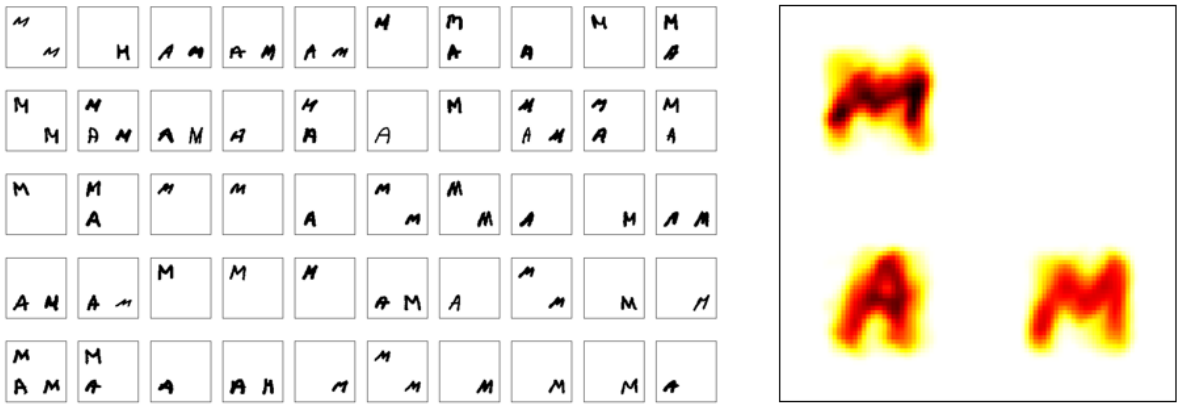


Figure 2.10: *(left)* UWB for a dataset of letters M-A-M built in the same logic than fig. 2.8 with 50 figures: *(right)* resulting UWB with $\gamma = 0.01$, computed in 200 seconds using 10 processors.

Chapter 3

The constrained Wasserstein barycenter problem

This chapter extends the notion of Wasserstein barycenters in order to make the core optimization problem underlying scenario tree reduction more flexible, paving the way for further research.

Abstract This chapter presents two optimization methods to compute, subject to constraints, a Wasserstein barycenter (WB) of finitely many empirical probability measures. The new measure, denoted by constrained Wasserstein barycenter, extends the applicability of the standard WB to pre-required geometrical or statistical constraints. Our first approach is an extension of the Method of Averaged Marginals described in Chapter 2 to compute WBs subject to convex constraints. In the nonconvex setting, we propose an optimization model whose necessary optimality conditions are written as a linkage problem with non-elicitable monotonicity. To solve such a linkage problem, we combine the Progressive Decoupling Algorithm (Rockafellar, 2019) with Difference-of-Convex programming techniques. We give the mathematical properties of our approaches and evaluate their numerical performances in two applications, demonstrating both their computational efficiency and the practical relevance of constrained Wasserstein barycenters.

The main content of this chapter will appear in [84]: Mimouni, de Oliveira, Sempere, (2025). *On the Computation of Constrained Wasserstein Barycenters.* Pacific Journal of Optimization for a special issue dedicated to Professor R. Tyrrell Rockafellar.

3.1 Introduction

Due to its ability to aggregate and summarize probability measures while preserving spatial characteristics, the concept of Wasserstein barycenter has gained prominence across diverse applications, ranging from applied probability, passing through imaging to machine learning [27].

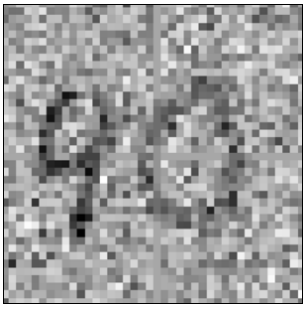
While the dedicated literature on numerical methods for computing WBs focuses mostly on the unconstrained setting, that is, μ can be freely chosen in the space $\mathcal{P}(\mathbb{R}^d)$, many practical situations require the target barycenter to satisfy certain constraints, ensuring it belongs to a predefined closed set $\mathfrak{X}$. Mathematically, the problem of computing a *constrained Wasserstein barycenter* (CWB) can be formulated as

$$\min_{\mu\in\mathcal{P}(\mathbb{R}^d)} \frac{1}{M}\sum_{m=1}^{M} W_2^2(\mu,\nu^m) \quad \text{s.t.} \quad \mu\in\mathfrak{X}\,. \tag{3.1}$$

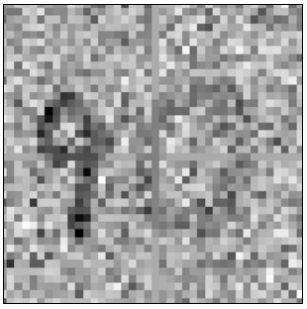
The additional constraint $\mu \in \mathfrak{X}$ is particularly relevant in applications where the barycenter must adhere to specific structural or operational requirements, or align with prior knowledge about the desired properties of μ. The set $\mathfrak{X}$ can influence the geometry of the barycenter, its statistical properties, its support, or its physical suitability. Hence, problem (3.1) offers greater modeling flexibility to tackle WB applications. For instance, the work [113] employs CWBs in image morphing applications. The authors consider variants of problem (3.1) with $\mathfrak{X}$ being manifolds, modeling sparsity or generative adversarial network (GAN)-based representations. They show that when compared with unconstrained WBs, model (3.1) provides superior results for tasks like natural image morphing, offering smooth, visually plausible transitions without introducing artifacts. By leveraging priors like GANs, [113] handles nonconvex constraints with a heuristic inspired by the ADMM algorithm.

Another application where a constrained barycenter is sought arises when summarizing images by restricting the total number of pixels that can have nonzero mass. In such an application, a black-and-white image can be associated with a probability measure: the pixels (positions) constitute the measure's support, while the intensity of each pixel represents the measure's (probability) mass. Figure 3.1c shows an unconstrained WB of the two top images 3.1a and 3.1b, computed by solving the unconstrained problem (2.4). Those images are of dimension 40×40, i.e., they have 1600 pixels. To compute a summarized picture with at most 80 pixels with non-zero mass, one may think of defining $\mathfrak{X}$ as being the set of 40×40 images with such a property and project the WB of Figure 3.1c onto $\mathfrak{X}$. Such a naive approach gives the image in Figure 3.1d, which does not keep the entire relevant information. On the other hand, by considering (a model for) the constrained problem (3.1) with such a set $\mathfrak{X}$ and applying the algorithm we present in Section 3.4, we get the image depicted in Figure 3.1e.

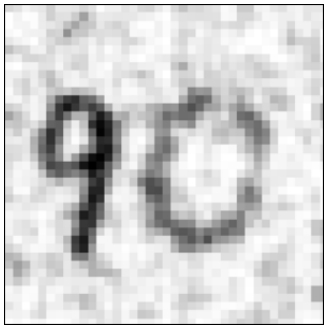
To the best of our knowledge, this is the first time that equation (3.1) is addressed in the general case. However, we note that it is not the only approach to modeling restrictions in WB applications. Some authors have studied constraints on the transport plans, usually to address specific applications, such as those arising in finance, martingale transport problems, and other domains (see, for instance, [10, 49, 57] and [94, §4.20 and §10.12]). In these contexts, methods based on iterative Bregman projections, such as those discussed in [94], have been adapted to handle the additional constraints by alternating projections. In either of these models, the convexity of the additional constraints evidently plays an important role in numerical optimization. But while assuming this convexity simplifies the optimization model, it also reduces its applicability in real-world tasks where nonconvex constraints are common. Furthermore, scalability and interpretability of solutions remain open challenges when dealing with high-dimensional or structured data.



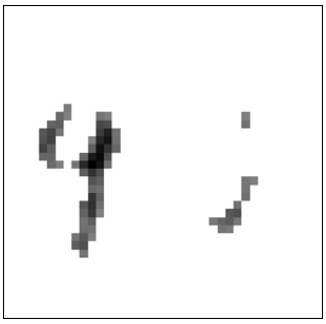
(a) Noisy image of a 90.



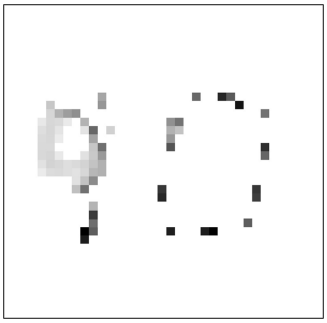
(b) Second noisy image of a 90.



(c) Unconstrained barycenter.



(d) Unconstrained barycenter projected onto $\mathfrak{X}$.



(e) Constrained barycenter.

Figure 3.1: Unconstrained WB, projection of the unconstrained WB, and sparse barycenter of the two noisy images 3.1a and 3.1b.

Contributions and organization This chapter presents two approaches to tackle constrained Wasserstein barycenter problems. Our first contribution on the computation of constrained WBs is the extension of the Method of Averaged Marginals (MAM) introduced in Chapter 2 to tackle problem (3.1). We show that our extension of MAM asymptotically computes an exact solution to (3.1) provided the constraints are convex. As the original method of [85], our algorithm copes with scalability issues and is memory efficient. In the nonconvex setting, our approach becomes an heuristic that works notably well in some applications, as evidenced by our numerical results.

To cope with nonconvexity in a mathematically sound approach, we propose a relaxed model for (3.1) based on the penalization of the squared distance from the nonconvex set. As we show in Section 3.4, our model consists of minimizing a nonsmooth Difference-of-Convex (DC) function over a linear subspace. Its necessary optimality conditions can be written as a linkage problem. However, since convexity cannot be elicited at any level, Rockafellar's compelling *Progressive Decoupling Algorithm* [101] (see also [102] and [117]) cannot be directly applied. Based on the recent work [45], we specialize the progressive decoupling strategy to the constrained WB setting to design an algorithm with convergence guarantees to solve such a linkage problem, computing thus points satisfying certain necessary optimality condition to our relaxed (penalized) model for (3.1). As a further contribution, we conduct experiments on several data sets from the literature to demonstrate the computational efficiency and accuracy of the new approaches.

This chapter is structured as follows. Section 3.2 provides some background material on Wasserstein barycenter problems. Section 3.3 extends the Method of Averaged Marginals (MAM) of [85] to tackle (3.1) when $\mathfrak{X}$ is convex.

Then, numerical results are shown where our approach is experimented with convex and nonconvex constraints. The precise case of nonconvex sets is addressed in Section 3.4, where we propose a relaxed model for (3.1) and present a progressive decoupling strategy with convergence guarantees. Finally, a numerical illustration comparing the approaches closes the work.

3.2 Background material

As developped in Chapter 2, the definition of the support of a measure ν^m entails that $\nu^m(\zeta_s^m) = q_s^m > 0$ for all $s = 1, \dots, S^m$. As computing a WB of M measures ν^m amounts to determine a new measure $\bar{\mu}$ solving (2.4), it turns out that such a task consists of choosing simultaneously a support $\texttt{supp}(\bar{\mu})$ and a probability vector $\bar{p}$ minimizing the (weighted) Wasserstein distance to all M measures. As for the decision on the support, Proposition 1 in [6] asserts that every solution $\bar{\mu}$ to (2.4) with $\alpha_m = \frac{1}{M}$ for all $m = 1, \dots, M$ has support satisfying the following key inclusion:

$$\begin{aligned} \texttt{supp}(\bar{\mu}) \subset \Xi &:= \left\{\xi^1, \dots, \xi^R\right\} \\ &:= \left\{ \frac{1}{M} \sum_{m=1}^{M} \zeta^m : \ \zeta^m \in \texttt{supp}(\nu^m), \ m = 1, \dots, M \right\}. \end{aligned} \tag{3.2}$$

Thanks to this result, we can work with the fixed set Ξ having finitely many R atoms[1] and optimize only with respect to the probability vector: once $\bar{p} \in \Delta_R$ is determined, we can recover a WB measure by setting

$$\bar{\mu} = \sum_{r \in \{j : \bar{p}_j > 0\}} \bar{p}_r \delta_{\xi_r}. \tag{3.3}$$

Because of (3.2), imposing a constraint of the type $\mu \in \mathfrak{X}$ can be done by restricting p in (2.8) in Chapter 2 to a certain set $X \subset \mathbb{R}^R$ related to $\mathfrak{X}$. In other words, in the empirical setting, the constrained WB problem (3.1) can be alternatively written as follows, for a set $X \subset \mathbb{R}^R$ associated $\mathfrak{X} \subset \mathcal{P}(\mathbb{R}^d)$:

$$\begin{cases} \min\limits_{p \in X, \ \pi \geq 0} & \displaystyle\sum_{m=1}^{M} \frac{1}{M} \sum_{r=1}^{R} \sum_{s=1}^{S^m} \|\xi_r - \zeta_s^m\|^2 \pi_{rs}^m \\ \text{s.t.} & (\pi^m)^\top \mathbf{1}_R = q^m, \quad m = 1, \dots, M \\ & \pi^m \mathbf{1}_{S^m} = p, \quad m = 1, \dots, M, \end{cases} \tag{3.4}$$

where $\mathbf{1}_R$ is the column vector of all ones of size R.
We highlight that this problem is solvable as long as X is closed and intersects the simplex Δ_R. Observe further that while (2.8) in Chapter 2 is always an LP, problem (3.4) can be a nonconvex nonlinear optimization problem depending on X. To give examples of how the set of constraints $\mathfrak{X}$ on the measure relates to the set of constraints X on the probability vector, suppose we impose Wasserstein barycenters to have expected value equal to a given $\bar{\xi} \in \mathbb{R}^d$. In this case,

$$\mathfrak{X} = \left\{\mu \in \mathcal{P}(\mathbb{R}^d) : \mathbb{E}_\mu[\xi] = \bar{\xi}\right\} \quad \text{and the corresponding set is} \quad X = \left\{ p \in \mathbb{R}^R : \sum_{r=1}^{R} p_r \xi_r = \bar{\xi} \right\}.$$

[1] The WB's support size R is determined in function of the number M of measures ν^m and their support sizes S^m, $m = 1, \dots, M$; see [6].

Such a setting finds applications, for instance, in scenario tree reduction where one might be interested in assigning probabilities to a smaller scenario tree in order to minimize the sum of Wasserstein distances while preserving, in every subtree issued by a node, certain theoretical expected value; see [47, §3.1] and [86]. If, instead, we require μ to have a support size of at most $n \geq 1$ atoms, then

$$\mathfrak{X} = \left\{\mu \in \mathcal{P}(\mathbb{R}^d) : |\texttt{supp}(\mu)| \leq n\right\} \quad \text{and the corresponding set is} \quad X = \left\{p \in \mathbb{R}^R : \|p\|_0 \leq n\right\},$$

where $\|p\|_0$ counts the number of nonzero components of the vector p. This is the setting considered in Figure 3.1e.

3.3 Constrained Wasserstein barycenters

In the unconstrained setting, the Method of Averaged Marginals (MAM) proposed in Chapter 2 solves the LP problem (2.8) in Chapter 2 by exploiting its particular structure and applying the Douglas-Rachford splitting method (DR) [50,53]. The resulting algorithm is memory efficient, can run in a deterministic or randomized fashion, copes with scalability issues, and has convergence guarantees. It updates transportation plans by projecting (in parallel) several vectors of dimension R onto sets of the form of $\Delta_R(\tau)$, with given scalar $\tau > 0$. This task can be performed *exactly* and efficiently using specialized methods [35]. Once the transportation plans are updated, its marginals p^m, $m = 1, \ldots, M$ are easily computed and an estimation of the probability vector yielding a WB is computed by averaging these marginals. We refer the interested reader to [85, §5] for a thorough discussion about the method and its convergence analysis. In what follows, our goal is to extend MAM to compute constrained WBs.

3.3.1 The method of averaged marginals for constrained WBs

This section features our first contribution: the extension of MAM to compute a solution $\bar{\mu}$ to problem (3.1). As discussed in Section 3.2, this amounts to compute a p-part solution $\bar{p}$ to problem (3.4) and recover $\bar{\mu}$ as in (3.3). To this end, we make the following assumption.

Assumption 1. *The set $X \subset \mathbb{R}^R$ in (3.4) is closed, and satisfies $X \cap \Delta_R \neq \emptyset$. Furthermore, the Euclidean projection onto it is convenient to execute.*

We recall that closeness of X and condition $X \cap \Delta_R \neq \emptyset$ are enough to ensure that (3.4) is solvable. Next, by denoting

$$c_{rs}^m := \frac{1}{M}\|\xi_r - \zeta_s^m\|^2 \;\; \forall r,\, s,\, m, \qquad \text{and inner product} \quad \langle c, \pi\rangle := \sum_{r,s,m} c_{rs}^m \pi_{rs}^m,$$

we drop the decision variable p in (3.4) and rewrite the problem in the following compact form:

$$\min_{\pi \in \mathcal{B}_X} \langle c, \pi\rangle \quad \text{s.t.} \quad \pi^m \in \Pi^m, \quad m = 1, \ldots, M, \tag{3.5a}$$

where

$$\Pi^m := \left\{\pi^m \geq 0 : (\pi^m)^\top \mathbf{1}_R = q^m\right\}, \; m = 1, \ldots, M, \tag{3.5b}$$

and

$$\mathcal{B}_X := \left\{\pi = (\pi^1, \ldots, \pi^M) : \pi^1 \mathbf{1}_{S^1} = \pi^2 \mathbf{1}_{S^2} = \cdots = \pi^M \mathbf{1}_{S^M} \in X\right\}. \tag{3.5c}$$

Thus, once problem (3.5) is solved, we can easily recover a p-solution to problem (3.4) and, as a consequence, a constrained WB measure $\bar{\mu}$.

To solve (3.5), we follow the lead of [85] and employ the DR algorithm, which asymptotically computes a solution by repeating the following steps, with $k = 0, 1, \ldots$, given initial point $\theta^0 = (\theta^{1,0}, \ldots, \theta^{M,0})$, and prox-parameter $\rho > 0$:

$$\left\{ \begin{array}{lcl} \pi^{k+1} & = & \texttt{Proj}_{\mathcal{B}_X}(\theta^k) \\ \hat{\pi}^{k+1} & = & \arg\min\limits_{\pi \in \Pi} \langle c, \pi \rangle + \dfrac{\rho}{2} \|\pi - (2\pi^{k+1} - \theta^k)\|^2 \\ \theta^{k+1} & = & \theta^k + \hat{\pi}^{k+1} - \pi^{k+1}, \end{array} \right. \tag{3.6}$$

with $\Pi := \Pi^1 \times \ldots \times \Pi^M$. Assumption 1 ensures that the functions above are proper, convex, lower-semicontinuous, and problem (3.5) is solvable. The following is a direct consequence of Theorem 25.6 and Corollary 27.4 of [8].

Theorem 5. *Suppose X is convex and Assumption 1 holds. Then the sequence $\{\theta^k\}$ produced by the DR algorithm (3.6) converges to a point $\bar{\theta}$, and the following holds: $\bar{\pi} := \texttt{Proj}_{\mathcal{B}_X}(\bar{\theta})$ solves (3.5), and $\{\pi^k\}$ and $\{\hat{\pi}^k\}$ converge to $\bar{\pi}$.*

The DR algorithm is attractive when the two first steps in (3.6) are convenient to execute, which is the case in our setting as seen in Chapter 2.

The following original result, which does not require X to be convex, shows that the first step in (3.6) is simple provided the projection onto X is convenient to execute.

Proposition 4. *Let $\theta = (\theta^1, \ldots, \theta^M) \in \mathbb{R}^{R \times (S^1 + \cdots + S^M)}$, $a_m := \frac{\frac{1}{S^m}}{\frac{1}{S^1} + \cdots + \frac{1}{S^M}}$, and $p^m := \theta^m \mathbf{1}_{S^m}$, $m = 1, \ldots, M$. Given a nonempty and closed set $X \subset \mathbb{R}^R$, let $p \in \texttt{Proj}_X(\sum_{m=1}^M a_m p^m)$ and $\mathcal{B}_X$ given in (3.5c). Then, an element $\pi \in \texttt{Proj}_{\mathcal{B}_X}(\theta)$ has the form*

$$\pi^m_{:s} = \theta^m_{:s} + \frac{(p - p^m)}{S^m}, \tag{3.7}$$

for all $m = 1, \ldots, M$ and $s = 1, \ldots, S^m$.

Proof. Given an arbitrary $w \in \mathbb{R}^R$, let us define the set

$$\mathcal{B}_w := \left\{ \pi = (\pi^1, \ldots, \pi^M) : \ \pi^1 \mathbf{1}_{S^1} = \pi^2 \mathbf{1}_{S^2} = \cdots = \pi^M \mathbf{1}_{S^M} = w \right\}.$$

Observe that $\mathcal{B}_w$ is nonempty[2] and $\mathcal{B}_X$ in (3.5c) can be written as $\mathcal{B}_X = \cup_{w \in X} \mathcal{B}_w$. Therefore, computing a point in $\texttt{Proj}_{\mathcal{B}_X}(\theta)$ can be done by solving

$$\min_{y \in \mathcal{B}_X} \frac{1}{2}\|y - \theta\|^2 = \min_{w \in X,\, y \in \mathcal{B}_w} \frac{1}{2}\|y - \theta\|^2 = \min_{w \in X} \left\{ \min_{y \in \mathcal{B}_w} \frac{1}{2}\|y - \theta\|^2 \right\}.$$

The inner problem above is nothing but the projection onto $\mathcal{B}_w$. It can be written as

$$z = \texttt{Proj}_{\mathcal{B}_w}(\theta) \ = \ \left\{ \begin{array}{ll} \arg\min\limits_{y} & \dfrac{1}{2} \sum\limits_{m=1}^{M} \|y^m - \theta^m\|^2 \\ \text{s.t.} & \sum_{s=1}^{S^m} y^m_{rs} - w_r = 0, \quad r = 1, \ldots, R,\ m = 1, \ldots, M. \end{array} \right.$$

[2] The plans $\pi^m_{:1} = w$ and $\pi^m_{:s} = 0$ for $s \neq 1$ compose a point π that belongs to $\mathcal{B}_w$.

Being a solvable strongly convex quadratic program problem, the existence of Lagrange multipliers is ensured. As a result, the optimality conditions for this problem read as

$$(z_{rs}^m - \theta_{rs}^m) + \lambda_r^m = 0, \ \forall\, r, s, m \tag{3.8}$$

$$\sum_{s=1}^{S^m} z_{rs}^m = w_r, \ \forall\, r, m. \tag{3.9}$$

Note that summing equation (3.8) over $s = 1, \ldots, S^m$, with r and m fixed, gives

$$\lambda_r^m = \frac{\sum_{s=1}^{S^m} \theta_{rs}^m - \sum_{s=1}^{S^m} z_{rs}^m}{S^m} = \frac{p_r^m - w_r}{S^m}, \ \forall\, r, m$$

where the last equality follows by (3.9) and definition of p^m. As a result, we conclude that $z = \texttt{Proj}_{\mathcal{B}_w}(\theta)$ is given by

$$z_{rs}^m = \theta_{rs}^m + \frac{w_r - p_r^m}{S^m}, \ \forall\, r, s, m. \tag{3.10}$$

Next, we show that when $w \in X$ is an element of $\texttt{Proj}_X(\sum_{m=1}^M a_m p^m)$, then z above belongs to the set $\texttt{Proj}_{\mathcal{B}_X}(\theta)$. Indeed,

$$\|z - \theta\|^2 = \sum_{m=1}^M \|z^m - \theta^m\|^2 = \sum_{m=1}^M S^m \|\frac{w - p^m}{S^m}\|^2 = \sum_{m=1}^M \frac{1}{S^m}\|w - p^m\|^2,$$

and thus

$$\begin{aligned}
\arg\min_{w \in X} \left\{ \min_{y \in \mathcal{B}_w} \frac{1}{2}\|y - \theta\|^2 \right\} &= \arg\min_{w \in X} \frac{1}{2} \sum_{m=1}^M \frac{1}{S^m}\|w - p^m\|^2 \\
&= \arg\min_{w \in X} \frac{1}{2} \sum_{m=1}^M \frac{\frac{1}{S^m}\|w - p^m\|^2}{\frac{1}{\sum_{j=1}^M \frac{1}{S^j}}} \\
&= \arg\min_{w \in X} \frac{1}{2} \sum_{m=1}^M a_m \|w - p^m\|^2 \\
&= \arg\min_{w \in X} \frac{1}{2} \sum_{m=1}^M \left\{ a_m \|w\|^2 - 2 a_m w^\top p^m + a_m \|p^m\|^2 \right\} \\
&= \arg\min_{w \in X} \frac{1}{2} \left\{ \|w\|^2 \sum_{m=1}^M a_m - 2 w^\top \left(\sum_{m=1}^M a_m p^m \right) \right\} \\
&= \texttt{Proj}_X \left(\sum_{m=1}^M a_m p^m \right),
\end{aligned}$$

because $\sum_{m=1}^M a_m = 1$. This shows that the minimum value of $\min_{w \in X} \min_{y \in \mathcal{B}_w} \|y - \theta\|^2$ is reached at z given in (3.10) and $w \in \texttt{Proj}_X(\sum_{m=1}^M a_m p^m)$. The proof is thus complete. □

Recall that this proposition does not assume convexity of X. Hence, whether convexity is present or not, projecting onto $\mathcal{B}_X \subset \mathbb{R}^{R \times (S^1 + \cdots + S^M)}$ is simple as long as the projection onto $X \subset \mathbb{R}^R$ is easy to perform. By relying on

Propositions 3 and 4, we now gather and simplify the three steps of the DR Algorithm 3.6 to provide our extension of the MAM algorithm of [85] to the constrained setting. We start by invoking Proposition 3 that allows us to decompose the second step of DR Algorithm 3.6 into $\sum_{m=1}^{M} S^m$ simple projections: every column of $\hat{\pi}^{m,k+1}$ is given by

$$\hat{\pi}^{m,k+1}_{:s} = \texttt{Proj}_{\Delta_R(q^m_s)}\Big(2\pi^{m,k+1}_{:s} - \theta^{m,k}_{:s} - \frac{1}{\rho}c^m_{:s}\Big) \;\forall\, s, m.$$

It follows from Proposition 4, with $p^k \in \texttt{Proj}_X(\sum_{m=1}^m a_m p^{m,k})$, and $p^{m,k} = \theta^{m,k}\mathbf{1}_{S^m}$, that $2\pi^{m,k+1}_{:s} - \theta^{m,k}_{:s} = 2(\theta^{m,k}_{:s} + \frac{p^k - p^{m,k}}{S^m}) - \theta^{m,k}_{:s} = \theta^{m,k}_{:s} + 2\frac{p^k - p^{m,k}}{S^m}$. Therefore,

$$\hat{\pi}^{m,k+1}_{:s} = \texttt{Proj}_{\Delta_R(q^m_s)}\Big(\theta^{m,k}_{:s} + 2\frac{p^k - p^{m,k}}{S^m} - \frac{1}{\rho}c^m_{:s}\Big) \;\forall\, s, m.$$

Furthermore, the step $\theta^{k+1} = \theta^k + \hat{\pi}^{k+1} - \pi^{k+1}$ in the DR algorithm boils down to

$$\begin{aligned}\theta^{m,k+1}_{:s} &= \theta^{m,k}_{:s} + \texttt{Proj}_{\Delta_R(q^m_s)}\Big(\theta^{m,k}_{:s} + 2\frac{p^k - p^{m,k}}{S^m} - \frac{1}{\rho}c^m_{:s}\Big) - \theta^{m,k}_{:s} - \frac{p^k - p^{m,k}}{S^m} \\ &= \texttt{Proj}_{\Delta_R(q^m_s)}\Big(\theta^{m,k}_{:s} + 2\frac{p^k - p^{m,k}}{S^m} - \frac{1}{\rho}c^m_{:s}\Big) - \frac{p^k - p^{m,k}}{S^m},\; \forall\, s, m.\end{aligned}$$

Putting all together and removing the variables π^k and $\hat{\pi}^k$ we get the extension of MAM presented in Algorithm 2.

Algorithm 2 Method of Averaged Marginals for Constrained WBs

1: **Input**: M empirical probability measures $\nu^m \in \mathcal{P}(\mathbb{R}^d)$, initial plan $\theta^m \in \mathbb{R}^{R\times S^m}$, set X of constraints, and a scalar $\rho > 0$

2: Define $c^m_{rs} := \frac{1}{M}\|\xi_r - \zeta^m_s\|^2$, for all $r = 1,\dots,R$, $s = 1,\dots,S^m$ and $m = 1,\dots,M$
3: Set $a_m := (\frac{1}{S^m})/(\sum_{j=1}^M \frac{1}{S^j})$ and $p^m_r := \sum_{s=1}^{S^m} \theta^m_{r,s}$, $m = 1,\dots,M \quad r = 1,\dots,R$

4: **while** not converged **do**

5: $\quad p \leftarrow \texttt{Proj}_X(\sum_{m=1}^M a_m p^m)$

6: $\quad$ **for** $m = 1,\dots,M$ **do**
7: $\quad\quad$ **for** $s = 1,\dots,S^m$ **do**
8: $\quad\quad\quad \theta^m_{r,s} \leftarrow \texttt{Proj}_{\Delta_R(q^m_s)}\Big(\theta^m_{r,s} + 2\frac{p - p^m}{S^m} - \frac{1}{\rho}c^m_{r,s}\Big) - \frac{p-p^m}{S^m} \quad r = 1,\dots,R$
9: $\quad\quad$ **end for**
10: $\quad\quad p^m_r \leftarrow \sum_{s=1}^{S^m} \theta^m_{r,s} \quad r = 1,\dots,R$
11: $\quad$ **end for**

12: **end while**
13: **return** p

A possible manner to stop Algorithm 2 is when the barycenter estimate p stabilizes. As commented in [85, §5], this should be seen as a heuristic stopping test. In the next result, we index the variables p and θ in Algorithm 2 by k, which represents a pass between lines 4 and 12.

Theorem 6. *Suppose X is convex and Assumption 1 holds. Then the sequence $\{(p^k, \theta^k)\}$ produced by the Algorithm 2 converges to a point $(\bar{p}, \bar{\theta})$, with $(\bar{p}, \texttt{Proj}_{\mathcal{B}_X}(\bar{\theta}))$ solving (3.4).*

Proof. As presented, Algorithm 2 is the DR algorithm (3.6) applied to problem (3.5). Recall that X is convex and closed. As $X \cap \Delta_R \neq \emptyset$ in view of Assumption 1, problem (3.5) has a solution, and thus Theorem 5 ensures that: $\{\theta^k\}$ converges to a point $\bar{\theta}$, with $\texttt{Proj}_{\mathcal{B}_X}(\bar{\theta})$ a solution to (3.5). Proposition 4, with the additional assumption of convexity of X, asserts that

$$\bar{\pi}^m_{:s} = \bar{\theta}^m_{:s} + \frac{\bar{p} - \bar{p}^m}{S^m} \; \forall \, s, m,$$

solves (3.5), where $\bar{p}^m = \sum_{s=1}^{S^m} \bar{\theta}^m_{:s}$. As $\sum_{s=1}^{S^m} \bar{\pi}^m_{:s} = \bar{p}$, for all m, the relation between problems (3.5) ensures that $(\bar{p}, \bar{\pi})$ solves (3.4). □

Remark 3. *If $X = \mathbb{R}^R$, then Algorithm 1 boils down to the Method of Averaged Marginals of [85]. As in that paper, we can also randomize our algorithm by performing, at every iteration, the projections on line 8 only for a single measure m chosen at random. In this randomized setting, the convergence results above hold almost surely. This follows directly from the analysis in [85, Thm. 5.5].*

Remark 4. *Algorithm 2 is an adaptation of the classical Douglas-Rachford splitting method [50, 77]. This algorithm is known to exhibit a general convergence rate of $\mathcal{O}(1/k)$, where k denotes the number of iterations performed [62]. Equivalently, it requires approximately $\mathcal{O}(1/\epsilon)$ iterations to compute an ϵ-accurate solution to problem (3.4).*

While Algorithm 2 provides an exact manner for computing a constrained Wasserstein barycenter under Assumption 1, its simplicity and decentralized nature encourage its application in the nonconvex setting. The key insight is that if one can efficiently compute projections onto X, our approach could be (heuristically) applied to problem (3.5) even when X is nonconvex. The following section provides numerical insights into what can be obtained by Algorithm 2 in the convex and nonconvex settings.

3.3.2 Some numerical insights

This subsection presents some practical cases of why constrained WBs are worth considering. We compare unconstrained WB, convex and nonconvex constrained WBs, all computed by Algorithm 2 with three different choices for X (being the first choice $X = \mathbb{R}^R$). In the first example, we consider a simple case of a transportation problem with limited storage locations. In the second instance, a toy problem of image processing is presented. The goal is to obtain a sparse barycenter that accurately represents the initial data. Reproductible code is available at `https://github.com/dan-mim/Constrained-Optimal-Transport` [82].

3.3.2.1 Demand and location storage localization

We consider a localization problem for demand and storage optimization. The dataset comprises demand maps of a certain product for Paris over a 12-month period (one map per month). Figure 3.2 shows a sample of nine out of the twelve months, with each map illustrating the aggregated product demand in red. The objective is to determine optimal storage locations to minimize distribution costs.

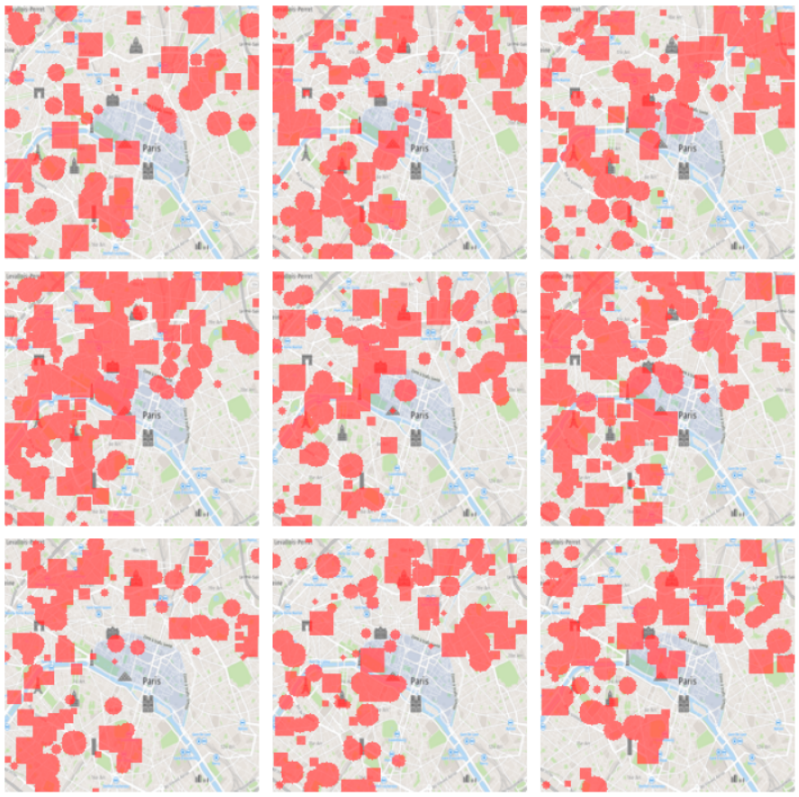
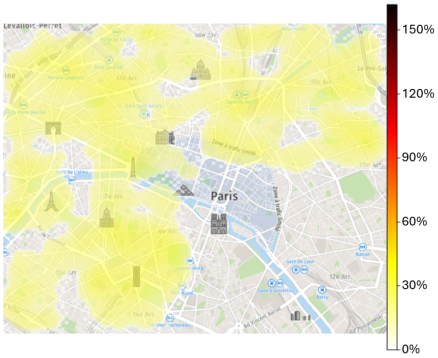


Figure 3.2: Sample of 9 (out of 12) months of collected demand data: the aggregated product demand is represented in red.

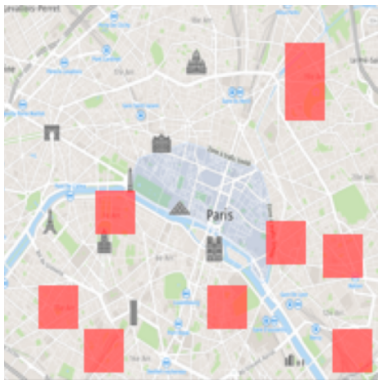
First, let us compute an unconstrained Wasserstein barycenter of the demand maps. Such a barycenter, presented in Figure 3.3a, suggests the need to rent 5726 storage facilities, which is impractical due to high costs in certain areas. Therefore, we restrict the storage to eight affordable locations $\mathcal{L}_\ell$ (marked in red in Figure 3.3b) with capacities $u_r > 0$ for all $r \in \left\{j = 1, \ldots, R : \xi^j \in \cup_{\ell=1,\ldots,8} \mathcal{L}_\ell\right\}$. As such restrictions yield probability (in this case demand) $p_r = 0$ for locations ξ^r outside the locations $\mathcal{L}_\ell$, the problem can be recast as a smaller Wasserstein problem with bound constraints given by the convex set:

$$X = \left\{ p : p_r \leq u_r, \quad \forall r \text{ s.t. } \xi^r \in \bigcup_{\ell=1,\ldots,8} \mathcal{L}_\ell \right\}.$$

Projecting the unconstrained barycenter of Figure 3.3a (probability vector p^u) onto the affordable areas depicted in Figure 3.3b results in the map shown in Figure 3.4a. Unfortunately, this projection violates the probabilistic nature of the barycenter, as the projected solution satisfies only 7% of the total demand: $\texttt{Proj}_X(p^u)^\top \mathbf{1}_R = 0.07$. To address this mismatch, we integrate the affordability constraint directly into the optimization problem to define problem (3.4), whose solution computed by Algorithm 2 is shown in Figure 3.4b. This solution achieves 100% ($\sum_{r=1}^{R} p_r = 1$) demand fulfillment while requiring only 625 storage facilities. Notably, it identifies new high-utilization locations (e.g., three sites in the bottom-right corner with approximately 40% occupancy).

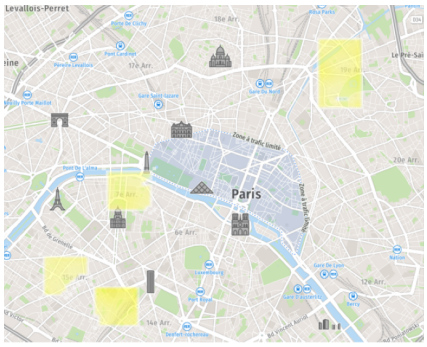


(a) Unconstrained WB with storage occupancy colorbar: 100% accounts for maximal u_r capacity being reached.

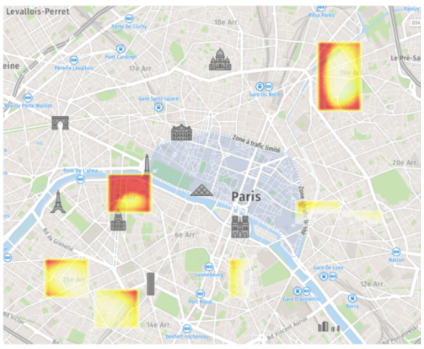


(b) Affordable areas are shown in red.

Figure 3.3: Unconstrained Barycenter of the demand maps and affordable storage locations.



(a) Unconstrained WB projected onto the set X.



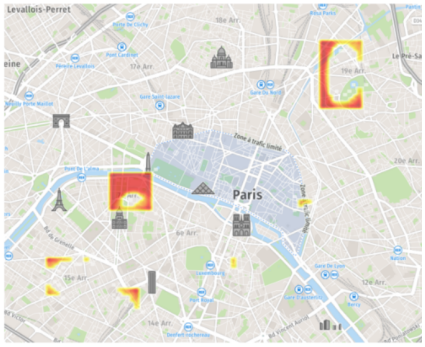
(b) Constrained WB computed by Algorithm 2.

Figure 3.4: Projected (unconstrained) WB versus constrained WB.

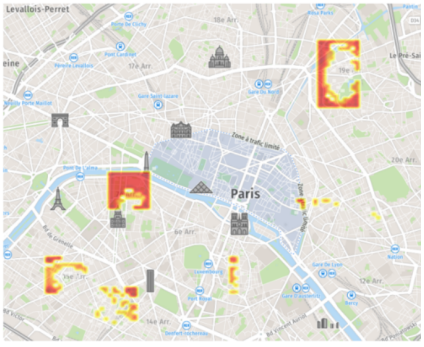
The (convex) constrained barycenter depicted in Figure 3.4b requires few storages in some locations $\mathcal{L}_\ell$. To maximize profitability, we introduce a nonconvex constraint that mandates a minimum storage utilization of 40%. The new constraint set is as follows:

$$X = \left\{ p : \, p_r \in \{0\} \cup [0.4u_r, u_r], \quad \forall r \text{ s.t. } \xi^r \in \bigcup_{\ell=1,\dots,8} \mathcal{L}_\ell \right\}.$$

Projecting the computed (convex constrained) barycenter onto this nonconvex set results in the allocation shown in Figure 3.5a, fulfilling only 64% of the demand (the projected barycenter is not a probability measure). However, integrating this nonconvex constraint directly into the optimization problem yields a point (Figure 3.5b) that satisfies 100% of the demand using only 308 storage facilities, compared to the 625 required previously.



(a) Convex constrained WB projected onto the nonconvex set X.



(b) Nonconvex constrained barycenter computed by Algorithm 2.

Figure 3.5: Projected convex constrained WB versus nonconvex constrained WB.

In summary, in this application, integrating constraints directly into the optimization process consistently produces better results than applying constraints post hoc. By incorporating both convex and nonconvex constraints, we achieve a practical solution that balances demand fulfillment and storage cost efficiency. We remark that for the dataset composed by 12 images of size 100×100, Algorithm 2 computed each one of the above (unconstrained, convex and nonconvex constrained) WBs within a couple of seconds, with none taking significantly longer than the others.

3.3.3 Sparse barycenter to a set of images

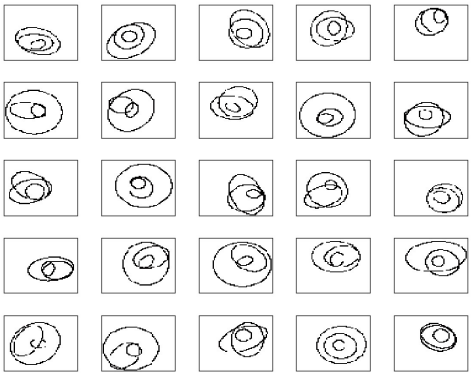
This subsection continues the discussion on the experiments presented in Figure 3.5 and evaluates Algorithm 2 in the context of nonconvex, constrained Wasserstein barycenter problems. We consider two test cases. In the first case, we show that our algorithm performs effectively, even though it is a heuristic in the nonconvex setting. The second test case illustrates a situation where the outcome produced by our approach is not satisfactory. In both cases, we work with the nonconvex set forcing sparsity:

$$X_{\mathtt{n}} = \{p : \|p\|_0 \leq \mathtt{n}\}\,.$$

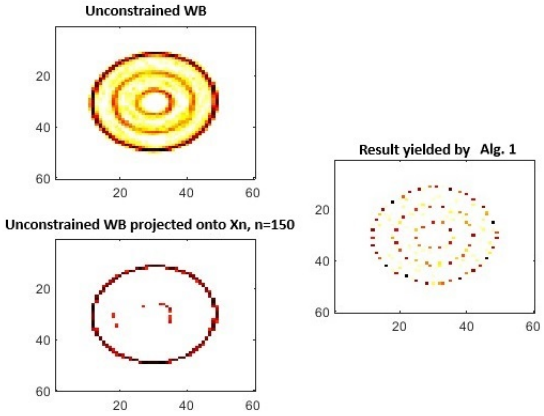
Here, $\mathtt{n}$ is a given natural number.
This application naturally arises when working with large datasets: the goal is not only to summarize the dataset with a Wasserstein barycenter, but also to compress the data to enable faster transmission and reduced memory storage. Similarly when dealing with scenario trees, we want to obtain a scenario tree that represents well the original one but with a smaller structure.

Ellipses: We consider a sample of $M = 100$ images of size $R = 60 \times 60$ with three nested ellipses. Figure 3.6a shows 25 of these images. A naive strategy to obtain a sparse image summarizing the dataset is first computing a unconstrained WB, and then projecting it onto $X_{\mathtt{n}}$. This is exemplified in Figure 3.6b (the two leftmost images) with $\mathtt{n} = 150$. This strategy is clearly unsatisfactory. On the other hand, Algorithm 2 applied with $X = X_{150}$ provides the rightmost image in Figure 3.6b, which clearly depicts three nested ellipses. In this test, problem (3.5)

has the order of 10^7 decision variables, and we let Algorithm 2 run 5000 iterations, which took about five minutes on a personal computer.



(a) Sample of 25 out of $M = 100$ images.

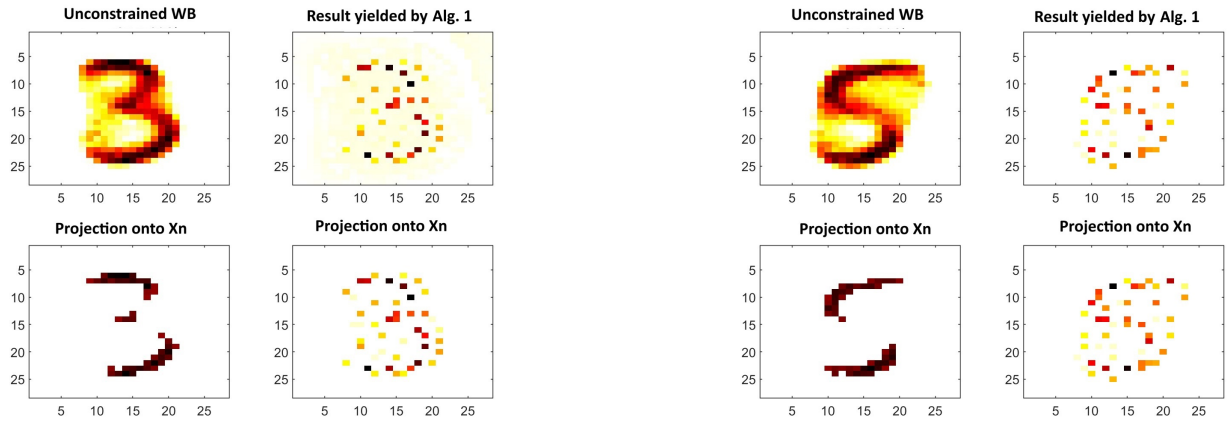


(b) Unconstrained WB, projection of the Unconstrained WB onto X_{150}, and image provided by Algorithm 2.

Figure 3.6: Sparse WB to a set of $M = 100$ images. The level of sparsity is chosen to be $\mathtt{n} = 150$.

MNIST: In this experiment, we use as initial input the well-known MNIST database, which includes $R = 28 \times 28$ images of handwritten numbers. By considering two samples of $M = 100$ images for the numbers three and five, we try to compute a sparse barycenter for each of the samples.

As in the previous example, we compare the computed unconstrained WB, its projection onto $X_{\mathtt{n}}$, and the point provided by Algorithm 2 with $X = X_{\mathtt{n}}$. In this experiment, problem (3.5) has the order of 10^6 decision variables, and we let Algorithm 2 run 5000 iterations, which took only a couple of minutes. While the unconstrained WB computed by MAM is meaningful, the sparse points provided by our approach fail to be probability vectors, and their quality in terms of visual meaning is poor. As shown in Figure 3.7, the projections of such points (including the unconstrained WB) onto $X_{\mathtt{n}}$ are not useful.



(a) Results for the number 3.

(b) Results for the number 5.

Figure 3.7: Unconstrained, sparse, and projection onto the nonconvex set $X_{\mathtt{n}}$, with $\mathtt{n} = 45$ for the digit three and $\mathtt{n} = 47$ for the digit five.

These experiments demonstrate that, while Algorithm 2 is guaranteed to produce meaningful results in the convex setting, its application as a heuristic in the nonconvex case does not consistently yield satisfactory outcomes. This observation highlights the necessity of the mathematically rigorous approach presented in the following section.

3.4 A progressive decoupling approach

Algorithm 2 is (asymptotically) exact when the set X is convex. However, the lack of convergence guarantees in the nonconvex case and the numerical illustration of the last section lead to question its use in some applications where a nonconvex set X is deemed indispensable. For this reason, we investigate in this section a penalized model for the nonconvex constrained Wasserstein problem (3.5) and a tailored solving methodology based on the *Progressive Decoupling Algorithm* (PDA) of Rockafellar [101]; see also [45, 102, 117].

Our initial, and unfruitful tentative, was to consider the following relaxed version of (3.5), with $\eta > 0$ a penalty parameter:

$$\min_{\pi} \langle c, \pi \rangle + \frac{\eta}{2} \mathtt{dist}^2_{\mathcal{B}_X}(\pi) \quad \text{s.t.} \quad \pi^m \in \Pi^m, \quad m = 1, \dots, M,$$

with $\mathtt{dist}^2_{\mathcal{B}_X}(\pi)$ the squared distance of π from $\mathcal{B}_X$. More precisely,

$$\mathtt{dist}^2_{\mathcal{B}_X}(\pi) = \min_{\theta \in \mathcal{B}_X} \frac{1}{2}\|\theta - \pi\|^2 = \frac{1}{2}\|\pi\|^2 - \max_{\theta \in \mathcal{B}_X} \left\{ \langle \theta, \pi \rangle - \frac{1}{2}\|\theta\|^2 \right\}$$

has a difference-of-convex (DC) structure: the two rightmost functions above are convex on variable π. As a result, the above penalized model is a DC programming problem [120, § 4.6] for which specialized algorithms exist [43, 108].

However, in our experiments, the results provided by such a DC model were not particularly appealing. This is why we propose to add the convex constraints

$$\pi \in \mathcal{B} := \mathcal{B}_{\mathbb{R}^R} \quad \text{(see Eq. (3.5c))}$$

to our DC model. The resulting (and more involving) optimization problem reads as follows:

$$\min_{\pi \in \mathcal{B}} \langle c, \pi \rangle + \frac{\eta}{2} \mathtt{dist}^2_{\mathcal{B}_X}(\pi) \quad \text{s.t.} \quad \pi^m \in \Pi^m, \quad m = 1, \dots, M. \tag{3.11}$$

Observe that this problem consists of minimizing a DC function over a linear subspace $\mathcal{B}$. Hence, any solution $\bar{\pi}$ to (3.11) is accompanied with a dual variable $\bar{y}$ solving the *linkage problem* [101]:

$$\text{find } \bar{\pi} \in \mathcal{B} \text{ and } \bar{y} \in \mathcal{B}^\perp \text{ such that } \ \bar{y} \in T(\bar{\pi}), \tag{3.12}$$

with $T(\pi) := \partial[\langle c, \pi \rangle + \sum_m \mathbf{i}_\Pi(\pi^m)] + \eta \partial^{\mathtt{C}} \mathtt{dist}^2_{\mathcal{B}_X}(\pi)$, and $\partial^{\mathtt{C}}$ denoting the Clarke subdifferential. (This linkage problem is nothing other than an alternative way to write the Lagrange system yielding a Clarke stationary point to (3.11).)

The work [101] investigates linkage problems and proposes the Progressive Decoupling Algorithm as a solving methodology. PDA solves the linkage problem should monotonicity of T be elicitable at a certain level. See also [117] and [102] for more details. However, being $\mathtt{dist}^2_{\mathcal{B}_X}(\pi) = \|\pi\|^2 - \big(2\langle \pi, \mathtt{Proj}_{\mathcal{B}_X}(\pi)\rangle - \|\mathtt{Proj}_{\mathcal{B}_X}(\pi)\|^2\big)$ a DC function, monotonicity of T above cannot be elicitable at any level, and thus the PDA of [101] is not directly applicable to our setting [45, § 2.5]. However, we can exploit the DC structure of problem (3.11) using the method proposed in [45]. To this end, let us write the objective function of (3.11) as

$$f(\pi) - h(\pi), \quad \text{with} \quad \begin{cases} f(\pi) := \langle c, \pi \rangle + \frac{\eta}{2}\|\pi\|^2 \\ h(\pi) := \eta \max\limits_{\theta \in \mathcal{B}_X} \left\{ \langle \theta, \pi \rangle - \frac{1}{2}\|\theta\|^2 \right\}. \end{cases}$$

Recall that $\mathcal{B}^\perp$ is the normal cone (at every point) of the linear subspace $\mathcal{B}$. Furthermore, note that the subdifferential set $\partial^{\mathtt{C}} h(\pi)$ coincides with the projection set $\mathrm{conv}(\eta \mathtt{Proj}_{\mathcal{B}_X}(\pi))$. The algorithm proposed in [45] linearizes h at a reference point π^{ℓ_k} (stability center) and defines a new stability center by inexactly solving the convex subproblem (see Algorithm 1 and Section 3.3 of [45]):

$$\min_{\pi \in \mathcal{B}} \ f(\pi) - \langle g^{\ell_k}, \pi \rangle + \frac{\mu}{2}\|\pi - \pi^{\ell_k}\|^2 \quad \text{s.t.} \quad \pi^m \in \Pi^m, \quad m = 1, \dots, M.$$

Here, $\mu > 0$ is a given parameter. Observe that such a subproblem is a quadratic variant of the unconstrained Wasserstein barycenter problem (2.8) in Chapter 2. To get around the practical inconvenience of solving difficult convex subproblems like this per iteration, the work [45] proposes to employ PDA with a safeguard permitting to stop the algorithm as soon as an incumbent point to (3.11) is found. In this work, the employed safeguard is a descent test accompanied by a penalty function associated with the constraints $\pi^m \in \Pi^m$. More specifically, we apply the PDA to the above subproblem to generate a sequence $\{\pi^k\}$ until a trial point π^{k+1} satisfying the following descent test is computed

$$F(\pi^{k+1}) \leq F(\pi^{\ell_k}) - \frac{\kappa}{2} v_k,$$

with $\kappa \in (0, \frac{1}{2})$,

$$F(\pi) := f(\pi) - h(\pi) + \mathtt{Penalty} \sum_m \mathtt{dist}_{\Pi^m}(\pi^m), \ \ \mathtt{Penalty} > 0,$$

and $v_k \geq 0$ defined in Algorithm 3 below. When such a descent test is satisfied, we halt PDA, set $\pi^{\ell_{k+1}} = \pi^{k+1}$, compute a new subgradient $g^{\ell_{k+1}} \in \eta \texttt{Proj}_{\mathcal{B}_X}(\pi^{\ell_{k+1}})$ to define the next subproblem, and repeat the process. Such a procedure can also be viewed as an inexact DC algorithm, where the convex subproblem is addressed by PDA but not solved to optimality. Our approach to tackle the nonconvex Wasserstein barycenter model (3.11) is presented in Algorithm 3.

Remark 5. *A few comments on Algorithm 3 are in order.*

- *Although steps 8, 15 and 25 all require equivalent loops over m and s, these loops cannot be merged, as they require the previous computation of p^{k+1} and $\bar{p}^{\ell_{k+1}}$ in steps 12 and 22, respectively. However, all of those can be run in parallel over m and s, as none of the steps depend on a previous iteration of the loop.*

- *If the descent test in step 20 holds for a given iterate $\ell_{k+1} = k+1$, the method requires computing a subgradient of h (equivalently a projection onto $\mathcal{B}_X$) at $\pi^{\ell_{k+1}}$. This step needs the partial sums over s of $\pi^{\ell_{k+1}}$. Note that the following relation holds*

$$\sum_{s=1}^{S^m} \pi_{:s}^{\ell_{k+1},m} = \sum_{s=1}^{S^m} \hat{\pi}_{:s}^{k,m} + (p^{k+1} - p^{k,m}) = p^{k+1},$$

  so the partial sums of $\pi^{\ell_{k+1}}$ (independent over m) are equal to p^{k+1}, which is already computed at line 12. By Proposition 4, projecting onto $\mathcal{B}_X$ is simple under Assumption 1.

- *In our experiments we have initialized y^0 as the vector of all zeros. We recall that if the projection onto $\mathcal{B}$ is easy, then the projection onto $\mathcal{B}^\perp$ is simple as well, since $\texttt{Proj}_{\mathcal{B}^\perp}(y) = \texttt{Proj}_{\mathcal{B}}(y) - y$.*

- *Algorithm 3 is a specialization of the method of [45] to our setting. In addition to the problem's structure exploitation, we have considered the simplifications discussed in Subsection 3.3 of [45].*

- *Note that Algorithm 3 needs three double loops, which increase proportionally with the size of the sample. Although these steps can be performed in parallel, the number of parallel calls a standard computer can handle is limited. In practice, these loops can significantly slow down the algorithm.*

Algorithm 3 PROGRESSIVE DECOUPLING ALGORITHM FOR NONCONVEX WB PROBLEMS

Input: M empirical probability measures $\nu^m \in \mathcal{P}(\mathbb{R}^d)$, initial primal and dual variables $\pi \in \mathcal{B}$ and $y^0 \in \mathcal{B}^\perp$, and scalars $\eta, \rho > 0$, $\kappa \in (0, \frac{1}{2})$, $\mu > 2\kappa$ and $\texttt{Penalty} > 0$

Define $c^m_{rs} := \frac{1}{M}\|\xi_r - \zeta^m_s\|^2$, for all $r = 1, \dots, R$, $s = 1, \dots, S^m$ and $m = 1, \dots, M$

Set $\ell_0 = 0$, $g^0 \in \eta \texttt{Proj}_{B_X}(\pi^0)$ and $a_m := (\frac{1}{S^m})/(\sum_{j=1}^M \frac{1}{S^j})$, $m = 1, \dots, M$

while not converged **do**
  $z^k \leftarrow c - y^k - g^{\ell_k} - \rho\pi^k - \mu\pi^{\ell_k}$
  for $m = 1, \dots, M$ **do**
    for $s = 1, \dots, S^m$ **do**
      $\hat{\pi}^{k,m}_{r,s} \leftarrow \texttt{Proj}_{\Delta(q^m_s)}\left[-\frac{1}{\eta+\rho} z^{k,m}_{rs}\right] \quad r = 1, \dots, R$ ▷ Projection onto the simplex
    end for
    $p^{k,m}_r \leftarrow \sum_{s=1}^{S^m} \hat{\pi}^{k,m}_{r,s} \quad r = 1, \dots, R$
  end for
  $p^{k+1} \leftarrow \sum_{m=1}^M a_m p^{k,m}$ ▷ Barycenter of the current iterate
  for $m = 1, \dots, M$ **do**
    for $s = 1, \dots, S^m$ **do**
      $\pi^{k+1,m}_{r,s} \leftarrow \hat{\pi}^{k,m}_{r,s} + \frac{p^{k+1} - p^{k,m}}{S^m} \quad r = 1, \dots, R$ ▷ Projection onto $\mathcal{B}$
    end for
  end for
  $y^{k+1} \leftarrow y^k - \rho(\hat{\pi}^k - \pi^k)$
  $v_k \leftarrow \max\{\|\pi^{k+1} - \pi^{\ell_k}\|^2, \|y^{k+1} - y^k\|^2, \|\pi^{k+1} - \pi^k\|^2\}$
  if $F(\pi^{k+1}) \le F(\pi^{\ell_k}) - \frac{\kappa}{2} v_k$ **then**
    $\ell_{k+1} \leftarrow k + 1$ ▷ Serious step
    $\bar{p}^{\ell_{k+1}} \leftarrow \texttt{Proj}_X(p^{\ell_{k+1}})$
    for $m = 1, \dots, M$ **do**
      for $s = 1, \dots, S^m$ **do**
        $g^{\ell_{k+1},m}_{r,s} \leftarrow \eta\left(\pi^{\ell_{k+1},m}_{r,s} + \frac{\bar{p}^{\ell_{k+1}} - p^{\ell_{k+1}}}{S^m}\right) \quad r = 1, \dots, R$ ▷ A subgradient
      end for
    end for
  else
    $\ell_{k+1} \leftarrow \ell_k$ ▷ Null step
  end if
end while
return $p^{\ell_{k+1}}$

Theorem 7 ([45, Thm. 3.1]). *Consider sequences $\{\pi^k\}_k$ and $\{\pi^{\ell_k}\}_k$ computed by Algorithm 3:*

1. *If only $\ell := \ell_k$ serious steps are performed, then $\lim_k \pi^k = \tilde{\pi}$ where $\tilde{\pi}$ solves the problem*

$$\min_{\pi \in \mathcal{B}} f(\pi) - [h(\pi^\ell) + \langle g^\ell, \pi - \pi^\ell \rangle] + \frac{\mu}{2}\|\pi - \pi^\ell\|^2 , \quad \text{s.t. } \pi^m \in \Pi^m \ \forall m = 1, \dots, M. \tag{3.13}$$

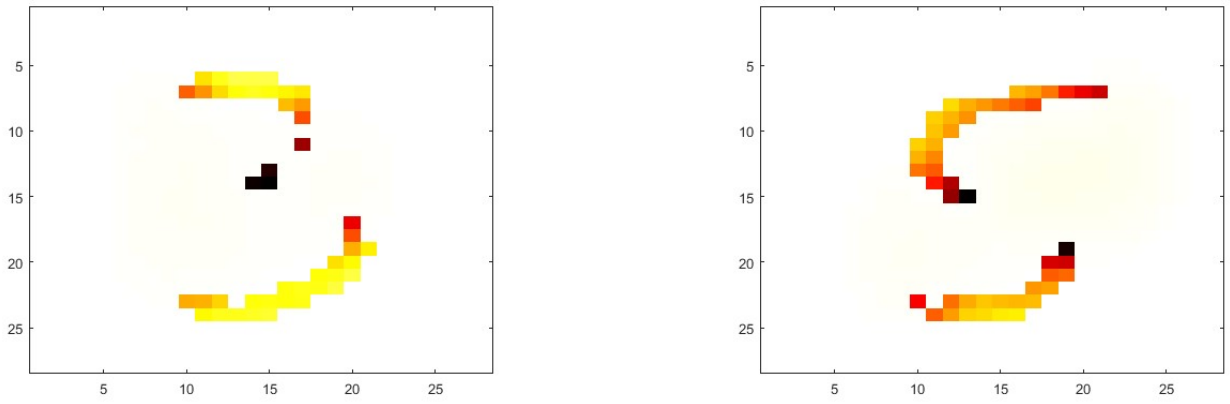
Moreover, if $\texttt{Penalty}$ *is large enough for* $\texttt{Penalty} \sum_m \texttt{dist}_{\Pi^m}$ *to be an exact penalty to problem (3.13), then $\tilde{\pi} = \pi^\ell$ solves the linkage problem (3.12).*

2. *If Algorithm 3 performs infinitely many serious steps, then every cluster point of* $\{\pi^{\ell_k}\}_k$ *solves the linkage problem* (3.12)

We highlight that the linkage problem (3.12) represents necessary optimality conditions to the penalized problem (3.11), which is a model for the WB problem (3.5).

Remark 6. *One of the conditions in Theorem 7 that ensures* π^ℓ *solves problem* (3.12) *is that the term* $\texttt{Penalty} \sum_m \texttt{dist}_{\Pi^m}$ *acts as an exact penalty. This property is known to hold when the parameter* `Penalty` *is sufficiently large [36, Prop. 9.1.1], although no explicit bound is known in our setting. Nevertheless, it is possible to detect, after many iterations of the algorithm, when* `Penalty` *is not large enough to ensure exact penalization. This issue is discussed in detail in [45, § 3.3].*

Sparse barycenter for MNIST: Let us return to the second test problem of Subsection 3.3.3, i.e, the MNIST dataset. As shown by Figure 3.7, Algorithm 2 fails to provide a satisfactory sparse WB when the nonconvex constraint $X_{\mathtt{n}}$ is considered in (3.5). By considering the model (3.11) and applying Algorithm 3 to that test problem we obtain the results depicted in Figure 3.8.



(a) Sparse WB for the number 3 with $\mathtt{n} = 45$.

(b) Sparse WB for the number 5 with $\mathtt{n} = 47$.

Figure 3.8: Examples of two sparse WB computed with Algorithm 3.

Note that Figure 3.8 shows more satisfactory results than those displayed in Figure 3.7. However, Algorithm 3 is significantly slower than Algorithm 2: Algorithm 3 performed 5000 iterations (for each example) in about thirty minutes. Furthermore, the memory usage of Algorithm 3 is not as efficient as that of Algorithm 2. These two drawbacks of Algorithm 3 should be addressed in future research.

Chapter 4

The scenario tree reduction problem

This chapter builds on the research developed in the previous chapters to propose a practical scenario tree reduction method that would enable the consideration of a large number of scenarios—by constructing a smaller tree that faithfully represents them—thereby enhancing the robustness of multistage scenario-based optimization models.

Abstract Scenario tree reduction techniques are essential for achieving a balance between an accurate representation of uncertainties and computational complexity when solving multistage stochastic programming problems. In the realm of available techniques, the Kovacevic and Pichler algorithm (Ann. Oper. Res., 2015 [70]) stands out for employing the nested distance, a metric for comparing multistage scenario trees. However, dealing with large-scale scenario trees can lead to a prohibitive computational burden due to the algorithm's requirement of solving several large-scale linear problems per iteration. This study concentrates on efficient approaches to solving such linear problems, recognizing that their solutions are Wasserstein barycenters of the tree nodes' probabilities on a given stage. We leverage optimal transport techniques to compute Wasserstein barycenters and significantly improve the computational performance of the Kovacevic and Pichler algorithm. Our boosted variants of this algorithm are benchmarked on several multistage scenario trees. Our experiments show that compared to the original scenario tree reduction algorithm, our variants can be eight times faster for reducing scenario trees with 8 stages, 78 125 scenarios, and 97 656 nodes.

The main content of Chapter 4 is under revision: Mimouni, Malisani, Zhu, de Oliveira, (2025) *Scenario Tree Reduction via Wasserstein Barycenters.* Annals of Operations Research [86].

4.1 Litterature review

Stochastic optimization techniques have proved essential in solving optimization problems in the presence of uncertainties driven by variables such as fluctuating prices, unpredictable demand, supply variations, resource availability,

and scheduling intricacies. The notion of stochastic programming is pioneered by Dantzig [42], and applications can be found in various sectors, including the financial industry [25, 54], supply chains [90], management science [29], energy economics [5, 12], electrical markets [52], hydro-thermal power systems [46] and maintenance of units [100].

Multistage stochastic programming, a class of stochastic optimization, often relies on scenario trees to represent the underlying stochastic process. In the quest for an accurate representation of uncertainties, large scenario trees are required. However, as the number of scenarios increases, so does the complexity of the problem and the computational effort regardless of the optimization method, whether it is based on scenario decomposition or stage decomposition [13]. Hence, finding a balance between an accurate representation of uncertainties and numerical tractability is of paramount importance in real-life applications modeled as multistage stochastic optimization problems.

In 2003, the influential work [52] introduced scenario reduction techniques based on the minimization of the Wasserstein distance between two scenario trees, pioneering the forward reduction and backward selection methodologies. Based on these ideas, [46] proposes to compute the Wasserstein distance at different nodes of a multistage scenario tree to design a scenario tree reduction algorithm. Differently, [74] formulates the scenario reduction problem as a mixed-integer linear programming problem. To decrease the computational effort when dealing with the Wasserstein distance, which can be formulated as a linear programming (LP) problem for finite sample of scenarios, subsequent works [69, 75] employ entropy-regularization schemes leveraged by the Sinkhorn–Knopp algorithm [115]. However, being based on the Wasserstein distance, these techniques ignore the filtration (structure) of multistage scenario trees. Neglecting the tree filtration potentially leads to deviations in solutions, because the solution of a multistage stochastic optimization problem is non-anticipative. As an attempt to cope with this shortcoming, the work [66] takes into consideration the so-called filtration distance, which relies on the tree structure. The stability results in that paper are the basis for the scenario tree reduction approach proposed in [65], which is a Wasserstein-based method that measures distances between nodes with the same parent, ultimately reducing pairs of nodes until a stopping criterion is met. However, this method encounters computational challenges, as it relies on solving NP-hard facility location problems. The work [31] proposes a scenario tree reduction algorithm that circumvents this difficulty by clustering tree nodes based on a new filtration distance and computes an approximation of the reduction problem. Scenario reduction strategies based on clustering are numerous in the literature, but often lack stability properties; see, for instance, [67, 128] and references therein.

A milestone in the field of scenario tree reduction was the introduction of the *nested distance* in [96], subsequently studied in [97]. The nested distance, also called *process distance*, offers a valuable framework for comparing multistage scenario trees as it considers the underlying filtrations. Leveraging the nested distance to guide scenario tree reduction enables control over the reduced tree's statistical quality and its impact on the objective value of the underlying multistage stochastic optimization problem. While it is relatively easy to calculate the nested distance between two given trees [98], finding a tree (with given filtration) that minimizes the nested distance is a much more challenging task. The reason is that the approximating tree's probabilities and support (i.e., the scenarios or outcomes) must be chosen to minimize the nested distance. This leads to a challenging optimization problem, which is large-scale and nonconvex. To tackle such a problem, Kovacevic and Pichler [70] introduced an algorithm that directly targets the minimization of the nested distance by alternating between probability and support optimizations. While the last task has an explicit solution (should the Euclidean norm be employed as a metric in the nested distance), optimizing the probabilities amounts to solving several large-scale LPs per stage. This represents the main bottleneck of the scenario tree reduction algorithm of [98], which can be partially avoided in some special cases. For instance, the work [12] provides a variant of the algorithm capable of handling large multistage scenario trees provided the stochastic process is *stage-wise independent*, a strong assumption we do not assume in this work.

Hence, for general stochastic processes, there is a clear need for more practical scenario tree reduction approaches based on the nested distance. This work contributes in this direction by boosting the Kovacevic and Pichler (KP) algorithm [98].

One of our contributions stems from recognizing that the algorithm's most time-consuming task, the probability optimization step, amounts to computing Wasserstein barycenters of the tree nodes' probabilities. In addition, we leverage optimal transport techniques to compute Wasserstein barycenters within the KP's algorithm by employing efficient and dedicated approaches such as the Iterative Bregmann Projection method (IBP) [15] and the Method of Averaged Marginals (MAM) [85], to attractively boost the computational performance for reducing scenario trees. As a third contribution, we benchmark our variants of the KP algorithm on real-life data and empirically show that they significantly outperform the baseline KP algorithm for large-scale multistage scenario trees. Our findings alleviate the algorithm's bottleneck, helping thus to elevate the KP approach among the top algorithmic choices for scenario tree reduction. We also emphasize the importance of the filtration initialization in the finding of a reduced tree.

The remainder of this chapter is organized as follows. Section 4.2 presents the Kovacevic and Pichler's approach and recalls the mathematical background for the nested distance. The barycentric approaches are introduced in Section 4.3, and the improved variant of the scenario tree reduction algorithm is developed. Some applications in Section 4.4 compare the new approach with the original algorithm by Kovacevic and Pichler, concluding with a speed-up method designed for real-life problems.

4.2 The Kovacevic and Pichler's approach

In [96], Pflug introduced the Nested Distance (ND), which is built on the Wasserstein distance, exploiting its structure and extending it to accommodate the specific characteristics of stochastic processes (see also [97]). It is a tailor-made measure for stochastic processes: it inherits the foundational properties of the Wasserstein distance, such as its sensitivity to local structure and robustness to outliers, while providing a more nuanced and specialized measure for comparing the similarities between different realizations of stochastic processes.

4.2.1 Scenario trees and notations

A T-period scenario tree ($\mathcal{N}$, A) is a discrete form of a random process – a family of random variables (see [70] for more). Figure 4.1 illustrates the concept of scenario tree. It is composed of (a set of) nodes $\mathcal{N}$ and edges A.

A node $m \in \mathcal{N}$ is a direct predecessor or parent of the node $n \in \mathcal{N}$, if $(m, n) \in A$, where (m, n) denotes the edge between the nodes m and n. This relation is embodied by the notation $m = n-$. Reciprocally, n is a direct successor (or child) of m and this set is denoted $m+$, such that $n \in m+$ if and only if $m = n-$. In the same vein, we denote the predecessors (or ancestors) of $n \in \mathcal{N}$ as $\mathcal{A}(n)$, the set of nodes for which there exists a path to n: for example $m_1 = m_2-$ and $m_2 = n-$ (equivalently $n \in m_2+$ and $m_2 \in m_1+$), then $m_1, m_2 \in \mathcal{A}(n)$. We consider only trees with a single root, denoted by 0, i.e., $0- = \emptyset$, but the methods developed here can be generalized for multi-root trees (forests). Nodes $n_T \in \mathcal{N}$ without successor nodes (i.e., $n_T+ = \emptyset$) are called leaf nodes. For every leaf node n_T there is a sequence $\xi_{n_T} := (n_0, \ldots, n_t, \ldots, n_T)$ where $n_0 \in \mathcal{A}(n_1)$, $n_1 \in \mathcal{A}(n_2)$ etc, from the root to the

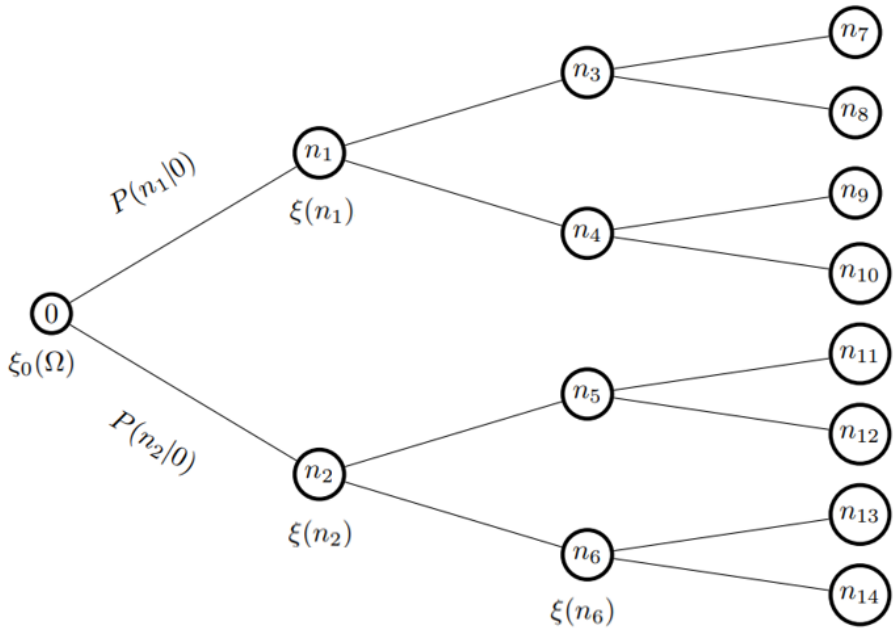


Figure 4.1: Scenario tree notations.

leaf node n_T composed by $T+1$ nodes. $\mathcal{F}_t$ is the filtration generated by the sigma-algebra of the family $(\xi_n)_{n\in\mathcal{N}_t}$ and we define $\mathcal{F} := (\mathcal{F}_t)_{t=1,\dots,T}$. The nodes bear a value named *quantizer*: $\xi : \mathcal{N} \mapsto \Xi$, we call $\xi(n)$ the quantizer of node n, where Ξ is a finite dimension metric space. We denote the probability, assigned to node n, by $P(n)$, that satisfies: $P(n) = \sum_{\tilde{n}\in n+} P(\tilde{n})$ for $n \in \mathcal{N}$ and $\sum_{n\in\mathcal{N}_T} P(n) = 1$. We denote conditional probability between successors by $P(n|m) = P(n)/P(m)$ for $m = n-$. Furthermore, using a distance $\mathtt{d} : \Xi^{t+1} \times \Xi^{t+1} \to \mathbb{R}$, we denote the distance between two nodes n_1, n_2 at stage t, as

$$\mathtt{d}_{n_1,n_2} := \sum_{n\in\xi_{n_1},\bar{n}\in\xi_{n_2},n \text{ and } \bar{n} \text{ at stage } t} \tilde{\mathtt{d}}(\xi(n),\xi(\bar{n})), \tag{4.1}$$

where $\tilde{\mathtt{d}}$ is a distance on Ξ.

4.2.2 Nested distance for trees

Using the tree notations introduced in Section 4.2.1, the process distance of order ι between two trees $\mathbf{P} := (\Xi^{T+1}, \mathcal{F}, P)$ and $\mathbf{P}' := (\Xi^{T+1}, \mathcal{F}', P')$ is the *discrete Nested Distance for Trees* (NDT). The transport mass between node $i \in \mathcal{N}_t$ and node $j \in \mathcal{N}'_t$ at stage $t \in \{1,\dots,T\}$, is noted $\pi_{i,j}$ or $\pi(i,j)$.

Definition 8 (Nested Distance for Trees)**.** *For $\iota \in [1,\infty)$, the process distance of order ι between $\mathbf{P}$ and $\mathbf{P}'$ is the ι^{th}-root of the optimal value of the following LP:*

$$\mathrm{ND}_\iota(\mathbf{P},\mathbf{P}') := \begin{cases} \min\limits_{\pi} & \sum\limits_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi(i,j)\mathtt{d}^\iota_{i,j} \\ s.t. & \sum_{\{j:n\in\mathcal{A}(j)\}} \pi(i,j|m,n) = P(i|m), \quad (m\in\mathcal{A}(i),n) \\ & \sum_{\{i:m\in\mathcal{A}(i)\}} \pi(i,j|m,n) = P'(j|n), \quad (n\in\mathcal{A}(j),m) \\ & \pi_{i,j}\geq 0 \ \textit{and} \ \sum_{i,j}\pi_{i,j} = 1. \end{cases} \tag{NDT}$$

Note that (NDT) is a generalization of the Wasserstein distance. Indeed, the transport plan π does not only respect the marginals imposed by P and P' but also respects the conditional marginals. These constraints embed the filtration in the definition of the distance between trees. Note that in the following, the term ND is used to refer to the value of (NDT).

This nuance is illustrated when deriving the W_2^2 or the ND_2 between the trees of Figure 4.2 (we use the Euclidean distance for $\tilde{\mathtt{d}}$).

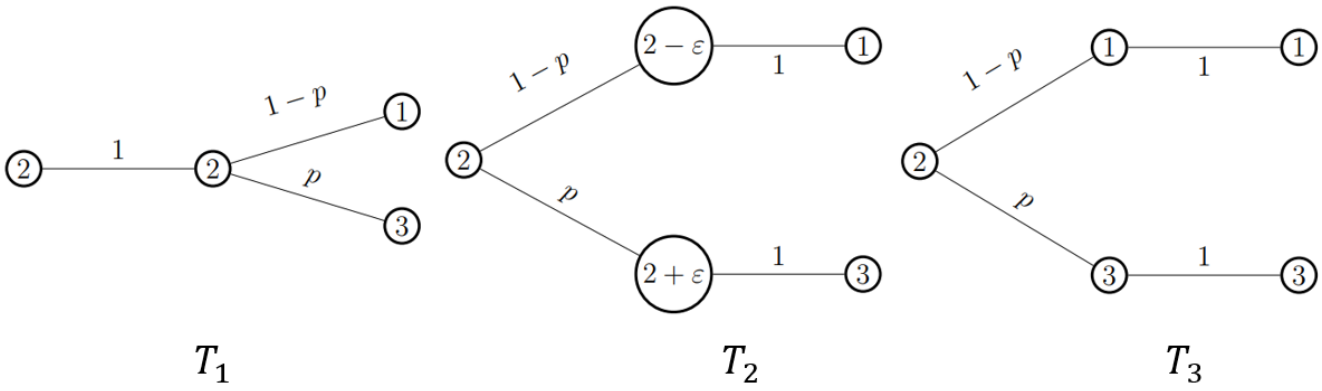


Figure 4.2: Three tree processes illustrating three different flows of information, taken from [95] and [70]. The Wasserstein distance of the first two trees vanishes, while the nested distance does not.

To compute the Wasserstein distance between two trees, we treat the trees as two scenarios. Here,

$$T_1: \quad \xi^1_{T_1} = (2,2,1), \quad \xi^2_{T_1} = (2,2,3),$$

and

$$T_2: \quad \xi^1_{T_2} = (2,2-\varepsilon,1), \quad \xi^2_{T_2} = (2,2+\varepsilon,3).$$

The squared Euclidean distance matrix between these scenarios is then

$$\begin{pmatrix} \varepsilon^2 & \varepsilon^2+4 \\ \varepsilon^2+4 & \varepsilon^2 \end{pmatrix}.$$

The minimization of eq. (2.8) obviously yields the following optimal transport plan:

$$\begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix}.$$

Therefore, we obtain

$$W_2(T_1, T_2) = \varepsilon.$$

When $\varepsilon \to 0$, the Wasserstein distance vanishes, even though the trees remain structurally different.

Now, let us compute the nested distance between T_1 and T_2. The $\mathtt{d}$-distance (see eq. (4.1)) between the roots is zero, while at the middle stage it is given by

$$\begin{pmatrix} \varepsilon^2 \\ \varepsilon^2 \end{pmatrix},$$

and at the final stage by

$$\begin{pmatrix} \varepsilon^2 & \varepsilon^2 + 4 \\ \varepsilon^2 + 4 & \varepsilon^2 \end{pmatrix}.$$

From (NDT), writing the constraints and using the conditional probabilities $\pi(i,j) = \pi(i,j \mid m,n)\,\pi(m,n)$, we obtain the transport plan at the middle stage:

$$\begin{pmatrix} 1-p \\ p \end{pmatrix},$$

and at the final stage:

$$\begin{pmatrix} (1-p)^2 & (1-p)p \\ (1-p)p & p^2 \end{pmatrix}.$$

Therefore,

$$\mathrm{ND}_2(T_1, T_2) = 2\varepsilon^2\big(1 + p(1-p)\big) + 8p(1-p).$$

In contrast to the Wasserstein distance, here ND_2 does not vanish when $\varepsilon \to 0$; it accounts for the structural difference between the two trees.

Similarly we can compute the distances for the other trees. We obtain $W_2(T_1, T_3) = 1$ and $W_2(T_2, T_3) = 1-\varepsilon$, which means that, as $\varepsilon \to 0$, the trees T_2 and T_3 become equidistant from T_1 in the sense of the Wasserstein distance, although they do not carry the same flow of information. This is not the case in the regard of the nested distance which captures the impact of the filtration: $\mathrm{ND}_2(T_1, T_3) = 2 + 10p(1-p)$, and $\mathrm{ND}_2(T_2, T_3) = 2(1-\varepsilon)^2$.

4.2.3 The Kovacevic and Pichler algorithm for scenario tree reduction

The scenario tree reduction mechanism involves solving structure-like (NDT) problems, between the original tree and the smaller one. The latter has given filtration (same number of stages but considerably fewer number of nodes than the original tree), initialized probabilities and quantizer values.

The scenario reduction optimization problem is nonconvex, due to the optimization of both quantizers and probabilities. The method in [70] operates a classical block coordinate optimization scheme. As illustrated in Figure 4.3, after a filtration is chosen, the first step is the optimization of the probabilities P' for fixed quantizers, and the

second step is the optimization of the quantizer values $\{\xi'(n) \in \Xi : n \in \mathcal{N}'\}$ for fixed probability. The latter step has an exact analytical solution in the Euclidean case, i.e. $\iota = 2$, and when $\tilde{\mathfrak{d}}$ in (4.1) is the Euclidean norm [70]. The probability optimization is more difficult because it requires solving multiple (potentially large-scale) LPs. In the remainder of this work, we will only develop the probability optimization; the quantizer optimization will only be recalled briefly in Algorithm 5.

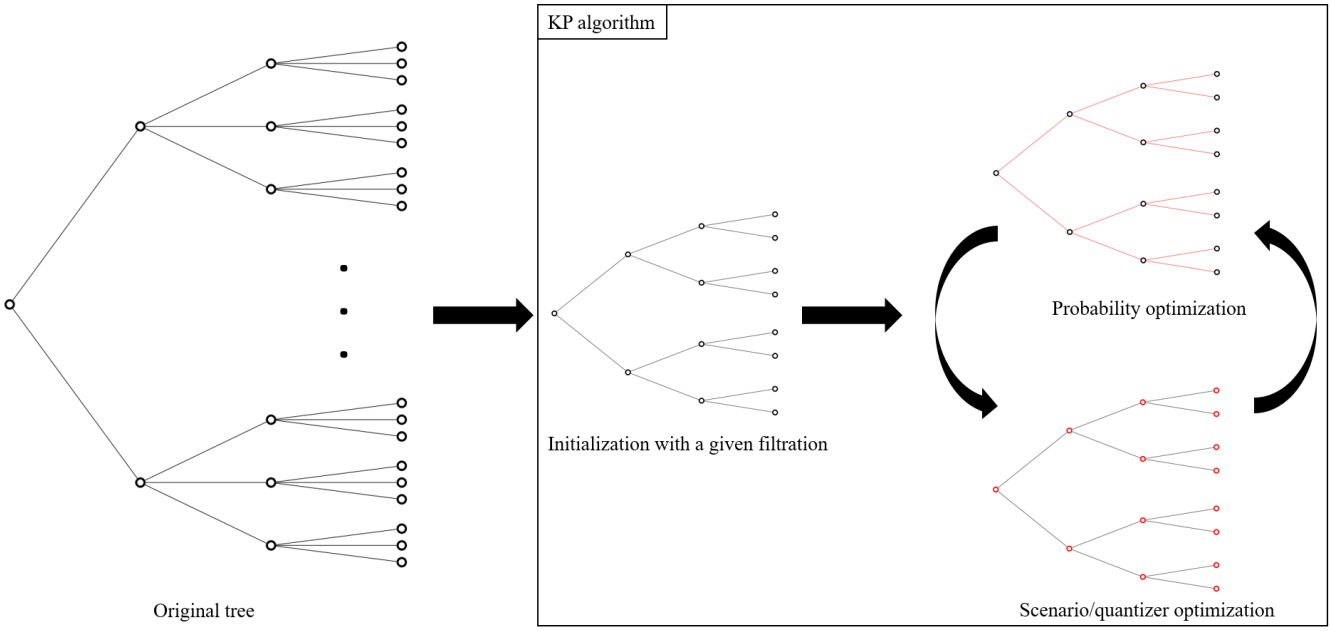


Figure 4.3: A general scheme of the Kovacevic and Pichler algorithm: to approximate a tree, a smaller tree with a given filtration is improved in order to minimize the nested distance with the original tree. The probabilities and the quantizers are alternatively optimized until convergence.

The probability optimization steps can be stated as follows: given the stochastic process quantizers $\{\xi'(n) \in \Xi : n \in \mathcal{N}'\}$ and structure of $(\mathcal{N}', A')$, we are looking for the optimal probability measure P' to approximate $\mathbf{P} := (\Xi^{T+1}, \mathcal{F}, P)$, regarding the nested distance. Inspecting formulation (NDT), it turns out that P' can be computed by jointly optimizing with respect to $\pi_{i,j}$ and $P'(j|j-)$. This leads to the following large nonconvex

optimization problem:

$$\begin{cases} \min\limits_{\pi,P'} & \sum\limits_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi(i,j)\mathtt{d}^\iota_{i,j} \\ \text{s.t.} & \sum_{j\in n+}\pi(i,j|m,n) = P(i|m), \quad (\forall m\in\mathcal{A}(i),n) \\ & \sum_{i\in m+}\pi(i,j|m,n) = P'(j|n), \quad (\forall n\in\mathcal{A}(j),m) \\ & \pi_{i,j}\geq 0 \text{ and } \sum_{i,j}\pi_{i,j}=1 \\ & P'(j|j-)\geq 0. \end{cases} \tag{4.2}$$

This is a bilinear problem, hence difficult to handle: there is a large number of decision variables and bilinear constraints (issued by the conditional probabilities in the second group of constraints, which involve the decision variables composing π and P'). Using the conditional probabilities $\pi(i,j) = \pi(i,j|m,n)\times\pi(m,n)$, we can derive the recursive formula (4.3a) below.
Let $\delta_\iota(m,n) := \sum_{i\in m+, j\in n+}\pi(i,j|m,n)\delta_\iota(i,j)$ for $m\in\mathcal{N}_t, n\in\mathcal{N}'_t$, and $\delta_\iota(i,j) = \mathtt{d}_\iota(\xi_i,\xi'_j)^\iota =: \mathtt{d}^\iota_{i,j}$ for the leaves i, j of the trees.

$$\begin{aligned} \sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T}\pi(i,j)\mathtt{d}^\iota_{i,j} &= \sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T}\pi(i,j)\delta_\iota(i,j) && \text{(4.3a)}\\ &= \sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T}\sum_{m\in i-, n\in j-}\pi(i,j|m,n)\pi(m,n)\delta_\iota(i,j) && \text{(4.3b)}\\ &= \sum_{n\in\mathcal{N}'_{T-1}}\sum_{m\in\mathcal{N}_{T-1}}\pi(m,n)\underbrace{\sum_{i\in m+, j\in n+}\pi(i,j|m,n)\delta_\iota(i,j)}_{\delta_\iota(m,n)} && \text{(4.3c)}\\ &= \sum_{n\in\mathcal{N}'_{T-1}}\sum_{m\in\mathcal{N}_{T-1}}\pi(m,n)\delta_\iota(m,n). && \text{(4.3d)} \end{aligned}$$

Note also $\delta_\iota(0,0) = \mathrm{ND}(\mathbf{P},\mathbf{P}')$, recursively. Thanks to this recursive formula given in [70], the problem can be split into recursive smaller problems for $m\in\mathcal{N}_t$ and $n\in\mathcal{N}'_t$: the conditional probability $\pi(\cdot,\cdot|m,n)$ is a solution to

$$\begin{cases} \min\limits_{\pi} & \sum_{m\in\mathcal{N}_t}\pi(m,n)\sum_{i\in m+, j\in n+}\pi(i,j|m,n)\delta_\iota(i,j) \\ \text{s.t.} & \sum_{j\in n+}\pi(i,j|m,n) = P(i|m), & (i\in m+) \\ & \sum_{i\in m+}\pi(i,j|m,n) = \sum_{i\in\tilde{m}+}\pi(i,j|\tilde{m},n), & (j\in n+ \text{ and } m,\tilde{m}\in\mathcal{N}_t) \\ & \pi(i,j|m,n)\geq 0. \end{cases} \tag{RP}$$

This reformulation of the problem is still bilinear, however, to overcome this difficulty, [70] proposes to fix $\pi(m,n)$ with the values computed from the previous iteration (or initialized at first iteration) giving rise to a LP approx-

imation. The authors have empirically shown that after few iterations of their algorithm the values assigned to $\pi(m,n)$ stabilize.

Despite this approximation, the problem is still challenging due to its huge dimensions. Thus, the method's main limitation is its computational burden that becomes prohibitive for large-scale scenario trees, as exemplified in Section 4.4. The reason is that the method requires solving potentially large-scale LPs as in (RP) repeatedly. We address the challenge by noticing that (RP) with fixed $\pi(m,n)$ for $m \in \mathcal{N}_t$ for a given n is a *Wassertein barycenter problem* (2.4), for which specialized and efficient algorithms exist (see for instance Chapter 2).

4.3 The probability optimization step is a Wasserstein barycenters problem

We invite the reader to refer to Chapter 2 for definitions of the Wasserstein distance (WD) and Wasserstein barycenters (2.4).

In what follows we present one of our contributions.

Proposition 5. *The recursive problem* (RP) *with fixed* $\pi(m,n)$ *for* $m \in \mathcal{N}_t$ *and a given* n *is a Wasserstein Barycenter problem.*

Proof. Let M empirical (discrete) measures ν^m having finite support sets:

$$\texttt{supp}(\nu^m) := \{\xi_1'^m, \dots, \xi_{S^m}'^m\} \quad \text{and} \quad \nu^m = \sum_{s=1}^{S^m} q_s^m \underline{\delta}_{\xi_s'^m}, \tag{4.4}$$

with $\underline{\delta}_u$ the Dirac unit mass on $u \in \mathbb{R}^d$ and $q^m \in \Delta_{S^m}$, $m = 1, \dots, M$.
A barycenter $\mu := \sum_{r=1}^{R} p_r \underline{\delta}_{\xi_r}$ with $p \in \Delta_R$ of the family $(\nu^m)_{m\in[1,M]}$ is a solution to (2.4).

Problem (2.4) (Chapter 2) can be equivalently formulated as a LP, where π^m for $m = 1, \dots, M$ denote the transport plan between the μ and the probability measures, see Section 2 of [85]:

$$\begin{cases} \min\limits_{p,\pi} & \alpha_1 \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^1} \mathtt{d}^1_{rs}\pi^1_{rs} & +\cdots+ & \alpha_M \sum\limits_{r=1}^{R}\sum\limits_{s=1}^{S^M} \mathtt{d}^M_{rs}\pi^M_{rs} & & \\ \text{s.t.} & \sum_{r=1}^{R} \pi^1_{rs} & & & = q^1_s, & s=1,\dots,S^1 \\ & & \ddots & & \vdots & \\ & & & \sum_{r=1}^{R} \pi^M_{rs} & = q^M_s, & s=1,\dots,S^M \\ & \sum_{s=1}^{S^1} \pi^1_{rs} & & & = p_r, & r=1,\dots,R \\ & & \ddots & & \vdots & \\ & & & \sum_{s=1}^{S^M} \pi^M_{rs} & = p_r, & r=1,\dots,R \\ & p\in\Delta_R, \pi^1\geq 0 & \cdots & \pi^M\geq 0, & & \end{cases} \tag{LP}$$

Let us now show that (RP) fits into this structure.
Let $\alpha^n_{m_i} := \pi(m_i, n)$ for $i = 1,\dots,M$, $M = |\mathcal{N}_t|$, and $(P'(j|n))_{j\in n+}$ an auxiliary vector of probabilities allowing to reformulate and couple the last group of constraints in (RP):$\sum_{i\in m+}\pi(i,j|m,n) = P'(j|n)$ for $j \in n+$, for all $m = m_1,\dots,m_M$. Given $t \in \{1,\dots,T\}$ and $n \in \mathcal{N}'_t$, problem (RP) reads as:

$$\begin{cases} \min\limits_{P',\pi} & \alpha^n_1 \sum\limits_{i\in m_1+, j\in n+} \pi(i,j|m_1,n)\delta_\iota(i,j) & +\cdots+ & \alpha^n_M \sum\limits_{i\in m_M+, j\in n+} \pi(i,j|m_M,n)\delta_\iota(i,j) & \\ \text{s.t.} & \sum_{j\in n+}\pi(i,j|m_1,n) & & = P(i|m_1), & (i\in m_1+) \\ & & \ddots & \vdots & \\ & & \sum_{j\in n+}\pi(i,j|m_M,n) & = P(i|m_M), & (i\in m_M+) \\ & \sum_{i\in m_1+}\pi(i,j|m_1,n) & & = P'(j|n), & (j\in n+) \\ & & \ddots & \vdots & \\ & & \sum_{i\in m_M+}\pi(i,j|m_M,n) & = P'(j|n), & (j\in n+) \\ & \sum_{j\in n+}P'(j|n)=1, \pi(i,j|m_1,n)\geq 0 & \cdots & \pi(i,j|m_M,n)\geq 0, & \end{cases} \tag{WB}$$

which is equivalent to

$$\min_{P'(.|n)\geq 0} \sum_{m\in\mathcal{N}_t} \alpha^n_m W_\iota((P'(j|n))_{j\in n+}, (P(i|m))_{i\in m+}) \quad \text{s.t.} \quad \sum_{j\in n+} P'(j|n) = 1,$$

i.e., a Wasserstein Barycenter problem.

□

To further illustrate our interpretation of problem (RP) as a Wassertein Barycenter problem, see Figure 4.4. The boxed subtree of the approximated (smaller) tree forms a probability measure with support (n_7, n_8) and probability

$(P(n_7|n_3), P(n_8|n_3))$; the probabilities of this subtree helping to minimize the ND is a barycenter of the (original tree's) subtrees with initial nodes m_3, m_4, m_5, m_6. The Wasserstein Barycenter $(P(n_7|n_3), P(n_8|n_3))$ is computed by solving (WB), the next (approximated tree's) subtree, that is the one issued by node n_4 is solved in the same manner etc. The probability $(P(n_9|n_4), P(n_{10}|n_4))$ are computed by solving another barycenter problem, but with different weights $\alpha_m^{n_4}$ (and costs) for $m \in \{m_3, m_4, m_5, m_6\}$ (see eq. (4.4)).

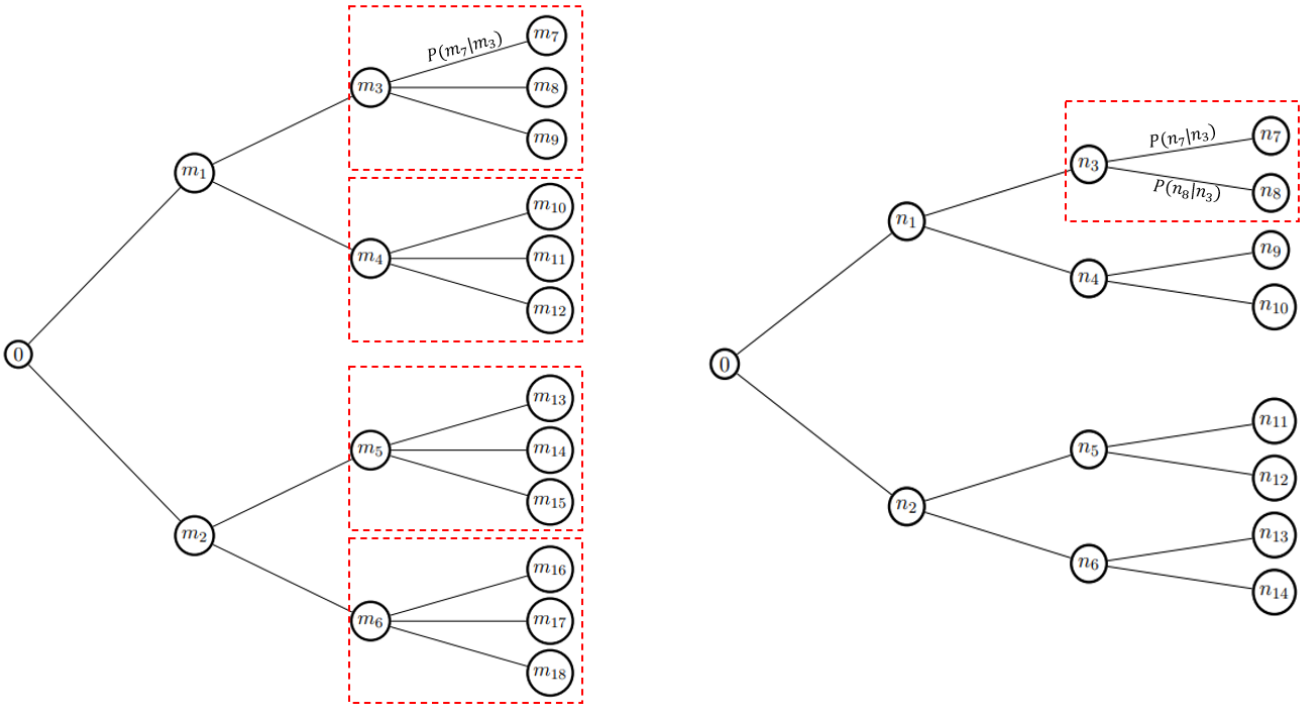


Figure 4.4: *(left)* Original tree, *(right)* Approximated tree. The probabilities $(P'(n_7|n_3), P'(n_8|n_3))$ are computed as the Wasserstein Barycenter of the set of (known) probabilities associated to the boxed subtrees on the left.

It is thus clear that several Wasserstein Barycenter problems must be solved at every iteration of the scenario tree reduction algorithm [70]. The faster the computation of such barycenters, the faster the Kovacevic and Pichler's algorithm.

4.3.1 Wasserstein barycenters techniques for scenario reduction

Problem from (WB) is an LP and could, in principle, be solved by LP solvers such as Gurobi, Cplex, HiGHs, and others. However, in real life applications of scenario tree reduction, problem (WB) (equivalently problem from (RP)) can have huge dimensions and be intractable by LP solvers, thus hindering the whole scenario tree reduction process. Many methods can be employed to tackle problem (WB) (for e.g. [15,38,131]). Note that these methods are

typically employed to retrieve a barycentric probability measure. However, in the context of this work, we instead focus on directly extracting the transport plans between the measures and the barycenter. We will further develop these concepts in the following sections. In the remainder of this chapter, we use two different WB algorithms, namely the Method of Averaged Marginals (MAM) [85] of Chapter 2 and the Iterative Bregmann Projection algorithm (IBP) [15]. In the following, we simplify the notation by denoting: $\pi(\cdot,\cdot|m_{\hat{m}},n) := \pi^{\hat{m}} \in \mathbb{R}^{R\times S^{\hat{m}}}$ for all $\hat{m}=1,\dots,M$, since n is fixed in (WB), and $P(\cdot|m)$ is denoted by $q^{\hat{m}}$.

4.3.1.1 The Method of Averaged Marginals (MAM)

MAM's main idea [85] is developed in Chapter 2 and its algorithm is detailed in Algorithm 1.

4.3.1.2 Computation of transport plans via regularized techniques

Utilizing efficient regularized methods enables us to quickly compute the transport plans required for the scenario tree reduction approach. For example, the *Iterative Bregman Projection* (IBP) [15] is a state-of-the-art technique, that applies regularization to the optimization problems presented in (WD) and (2.4).

Consider the entropic function:

$$E(\pi) := \sum_{r,s} \pi_{r,s}(\log(\pi_{r,s}) - 1), \tag{4.5}$$

with the convention $0\log(0) = 0$. This is a strongly convex function which assures that $E(\pi) \geq 0$ and $E(\pi) = 0$ if and only if $\pi = 0$. This function is employed to regularize the LP (WD), leading to the following nonlinear optimization problem:

$$\begin{aligned} {}^{\lambda}W(\mu,\nu) &:= \min_{\pi\in U(\mu,\nu)} \langle D,\pi\rangle + \frac{1}{\lambda}E(\pi) && \text{(4.6a)}\\ &= \min_{\pi\in U(\mu,\nu)} \frac{1}{\lambda}\sum_{r,s}\left(\lambda D_{r,s}\pi_{r,s} + \pi_{r,s}\log(\pi_{r,s}) - \pi_{r,s}\right) && \text{(4.6b)}\\ &= \min_{\pi\in U(\mu,\nu)} \frac{1}{\lambda}\,\mathrm{KL}(\pi|K) && \text{(4.6c)} \end{aligned}$$

Note that KL is the Kullback-Leibler divergence between $\pi, K \in \mathbb{R}^{R\times S}$ and $K_{r,s} > 0$ for all (r,s): $\mathrm{KL}(\pi|K) := \sum_{r,s}\pi_{r,s}\left(\log\left(\frac{\pi_{r,s}}{K_{r,s}}\right) - 1\right)$ (the precise definition of matrix K is given in the algorithm below).
By noticing that $p = \pi\mathbb{1}_R$, problem (WB) with ${}^{\lambda}W$ introduced in (4.6), instead of the classical Wasserstein distance, writes:

$$\min_{\substack{\pi^m \in U(\mu,\nu^m),\\ m=1,\dots,M}} \sum_{m=1}^{M} \alpha_m \,\mathrm{KL}(\pi^m|K^m) \tag{4.7}$$

Since problem (4.7) is strongly convex, it has a unique optimal solution. Following [15], the optimal coupling $\pi^m, m = 1, \dots, M$ can be derived after iterative KL projections onto the right and left constraints, embodied by the sets $U(\mu, \nu^m), m = 1, \dots, M$.

Algorithm 4 IBP ALGORITHM

Input: Given α_m for $m = 1, \dots, M$, $\lambda > 0$, initialize v^0 and u^0 with an arbitrary positive vector, for example $\mathbb{1}_S$
Initialize p^0, for example $\mathbb{1}_R/R$
Set $D^m \leftarrow \alpha_m \left(\delta_\iota(i,j)\right)_{(i,j)\in m+\times n+}$ and set $q^m = \left(P(i|m)\right)_{i\in m+}$
Define $K^m = e^{-\lambda D^m}$ for all $m = 1, \dots, M$
while not converged **do**

▷ Projections onto the constraints

for m=1,...,M **do**
$v^{m,k+1} = \frac{q^m}{(K^m)^T u^{m,k}}$
$u^{m,k+1} = \frac{p^{k+1}}{K^m v^{m,k+1}}$
end for

▷ Approximation of the barycenter

$p^{k+1} = \prod_{m=1}^{M} (K^m v^{m,k+1})^{\alpha_m}$
end while

return $\pi^m = \mathrm{diag}(u^m) K^m \mathrm{diag}(v^m)$ for all $m = 1, \dots, M$

Observe that all the algorithm's steps consist of matrix-vector multiplication, and are thus simple to execute. Note that p^{k+1} is the current estimate of the barycenter in Algorithm 4. The algorithm's drawback is its accuracy, which strongly depends on $\lambda > 0$. The greater is λ the closer is the solution of (4.7) to an exact solution of (WB). However, if λ is too large, the values of K diverge, leading to computational issues such as double-precision overflow errors.

4.3.2 Boosted Kovacevic and Pichler's algorithm

We rely on the previous subsections about Wasserstein barycenters computation methods to provide the following improved variant of the scenario tree reduction algorithm of [70]. In Algorithm 5, given a multistage scenario tree represented by $\mathbf{P} = (\Xi^{T+1}, \mathcal{F}, P)$ a smaller scenario tree $\mathbf{P'} = (\Xi^{T+1}, \mathcal{F}', P')$ is constructed by updating the quantizer values $\{\xi(n) \in \Xi : n \in \mathcal{N}\}$ and probability P', iteratively. The filtration $\mathcal{F}'$ is a data given to the algorithm. In other words, the number of scenarios and the structure of the reduced tree is an input data, and the algorithm seeks for the familly $\{\xi'(n) \in \Xi : n \in \mathcal{N}'\}$ and P' that minimizes the nested distance between $\mathbf{P}$ and $\mathbf{P'}$.

Some comments on Algorithm 5 are in order.

Initialization The probability $P^{'}$ and the scenario values $\{\xi'(n) \in \Xi : n \in \mathcal{N}'\}$ will be updated by the algorithm but the tree structure is fixed.

Algorithm 5 Scenario tree reduction via nested distance and Wasserstein barycenters

▷ Step 0: input

Let the original T-stage scenario tree $\mathbf{P} = (\Xi^{T+1}, \mathcal{F}, P)$ and a smaller scenario tree $\mathbf{P}^{'0} = (\Xi^{T+1}, \mathcal{F}', P^{'0})$ be given.

Compute a transport probabilities $\pi^0(i,j)$ between scenarios $(\xi_i)_{i\in\mathcal{N}_T}$ and $\left(\xi_j^{'0}\right)_{j\in\mathcal{N}_T'}$

Set $k \leftarrow 0$ and choose a tolerance $\texttt{Tol} > 0$

for $k = 1, 2, \ldots$ **do**

▷ **Step 1:** Improve the scenario values (quantizers)

  Set $\xi^{'k+1}(n_t) = \sum_{m\in\mathcal{N}_t} \frac{\pi^k(m,n_t)}{\sum_{i\in\mathcal{N}_t}\pi^k(i,n_t)}\xi_t(m)$ for all $n_t \in \mathcal{N}_t'$ for $t = 1, \ldots, T$

▷ **Step 2:** Improve the probabilities

  Set $\delta_\iota^{k+1}(i,j) \leftarrow \mathtt{d}(\xi_i, \xi_j)^\iota$ for all $i, j \in \mathcal{N}_T$

  for $t = T-1, \ldots, 0$ **do** ▷ Recursivity

    for all $n \in \mathcal{N}_t'$ **do** ▷ Wasserstein barycenters

      Set $\alpha_m^n \leftarrow \pi^k(m,n)$, $m \in \mathcal{N}_t$

      Use IBP, or MAM to compute $\pi^{k+1}(\cdot,\cdot|\cdot,n)$ solving (WB)

      Set $\delta_\iota^{k+1}(m,n) \leftarrow \sum_{i\in m+, j\in n+}\pi^{k+1}(i,j|m,n)\delta_\iota^{k+1}(i,j)$, $m \in \mathcal{N}_t$

    end for

  end for

▷ Build π^{k+1} the unconditional transport plan matrix

  Set $\pi^{k+1}(0,0) \leftarrow 1$

  for t=1,...,T **do**

    Compute $\pi^{k+1}(i,j) = \pi^{k+1}(i,j|m,n) \times \pi^{k+1}(m,n)$ for $i \in \mathcal{N}_t, j \in \mathcal{N}_t'$ and $m = i-$, $n = j-$

  end for

▷ **Step 3:** Stopping test

  if $\delta_\iota^k(0,0) - \delta_\iota^{k+1}(0,0) \leq \texttt{Tol}$ **then**

    Define $P'(n_T) = \sum_{m_T\in\mathcal{N}_T}\pi^{k+1}(m_T, n_T)$ for all $n_T \in \mathcal{N}_T'$ then $P'(n) = \sum_{j\in n+}P'(j)$ for all $n \in \mathcal{N}_t', t \neq T$

    Set $\mathrm{ND}_\iota(\mathbf{P}, \mathbf{P}') \leftarrow \delta_\iota^{k+1}(0,0)$

    Stop and return with the reduced tree $\mathbf{P}' = (\Xi^{T+1}, \mathcal{F}', P')$ and nested distance $\mathrm{ND}_\iota(\mathbf{P}, \mathbf{P}')$

  end if

end for

We will provide some initialization methods in Section 4.4.3.
Note that algorithm 5 is a straightforward operation since the initial transport plan only needs to respect the marginal constraints, therefore $\pi^0(i,j) = \frac{P(i|m)}{|n+|}$ for all $m, n, i \in m+, j \in n+$ is a sufficient initialization.

Quantizer optimizations Algorithm 5 only considers the Euclidean case ($\iota = 2$) in algorithm 5, but if $\iota \neq 2$, the quantizer optimization step boils down to a gradient descent (see [70] for details) and the algorithm is still applicable although slower.

Hyperparameters IBP relies on the use of a parameter $\lambda > 0$ chosen by the user [15, 39]. Such parameter has an impact on the result's accuracy. The MAM algorithm also requires setting a parameter, but it only impacts the convergence speed and can be determined with a sensitivity analysis [85].

Parallelization MAM is a parallelizable (and randomizable) algorithm, such a feature can also be leveraged in Algorithm 5. Algorithm 5 can also be treated in a parallel manner.

Stopping criteria The given stopping criteria is a heuristic: the algorithm terminates when the improvement of the nested distance between the two trees is below a certain level of tolerance `Tol`.

Convergence The algorithm leads to an improvement in each iteration. It should be kept in mind that the above algorithm is nothing but a heuristic, as it is a block-coordinate scheme seeking to minimize the nested distance by alternating minimization over scenarios and then with respect to probabilities.

4.4 Applications

This section illustrates the performance of Algorithm 5 and its behaviour when using different solvers to compute Wasserstein barycenters (i.e. solutions of (RP) and (WB)). All the following applications employ Algorithm 5 in the Euclidean case, though similar conclusions can be extended to other values of ι. If a standard LP solver is employed, then Algorithm 5 boils down to the setting of [70]. We ensure that the considered solvers attain the same level of precision when tackling the Wasserstein barycenter problems at Step 2 of Algorithm 5. Numerical experiments were conducted using 20 cores (*Intel(R) Xeon(R) Gold 5120 CPU*) and *Python 3.9*, using a *MPI* based parallelization when possible. The test problems and solvers' codes are available for download at the link `https://github.com/dan-mim/Nested_tree_reduction` [83].

4.4.1 Impact of the tree size

In what follows, we consider scenarios trees composed by 4 to 8 stages and 5 to 6 children per nodes, offering numbers of scenarios and spanning from hundreds to ten thousands nodes (see Table 4.1 for details). The quantizers of both trees are set randomly within the range $[-10, 10]$. In our preliminary tests, the reduced tree is always binary.

We consider the following variants of the algorithm, by varying the solver used to compute the Wasserstein Barycenter in Step 2 of Algorithm 5:

- *LP*: Algorithm 5 employing the LP solver HiGHS;
- *MAM*: Algorithm 5 employing the *Method of Averaged Marginals* [85];

- *IBP*: Algorithm 5 employing IBP, inspired from the code of G. Peyré [92]. This algorithm relies on the tune of an hyperparameter that has been preset to guarantee its best efficiency and convergence.

All variants use `Tol` $= 0.1$ as a tolerance for the stopping test. It has been empirically verified that results do not improve significantly if we decrease this tolerance further.

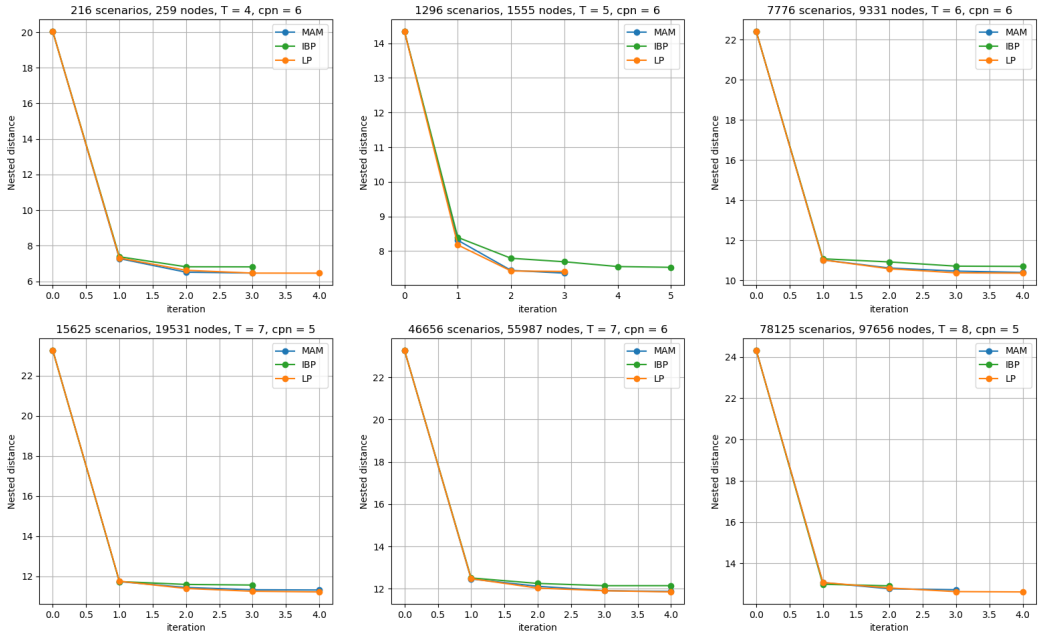


Figure 4.5: Evolution of the Nested Distance along the reduction iterations for different initial tree sizes. The final reduced tree has always two children per node.

Figure 4.5 shows that the exact ND between the original tree and the approximate one, iteratively decreases, no matter the variant of Algorithm 5. All variants are initialized with the same tree, generated randomly (probabilities and scenarios). Note that the initial ND is always at least halved after the reduction. This emphasizes how important is the use of a reduction method. Within this scale, one can see that every variant converges to approximately the same precision, although not necessarily to the same reduced scenario tree (because different solvers compute different optimal transportation plans, impacting the construction of scenarios composing the reduced tree).

Even though the ND decreases with all variants, it seems faster (in terms of number of iterations) when using LP or MAM. Figure 4.6 shows that the IBP algorithm tends to reach a less precise plateau. This is due to the core of the IBP method, which is an inexact algorithm. MAM being an exact algorithm for solving (WB), it naturally follows the lead of LP. The slight differences between the variants LP and MAM can be explained by the fact that the general problem is nonconvex and different optimal transportation plans computed by different solvers can lead to different optimization paths that result in different reduced scenario trees. Therefore, depending on the employed solvers and the initialization, the approximated trees could be different while still having close ND with the original tree.

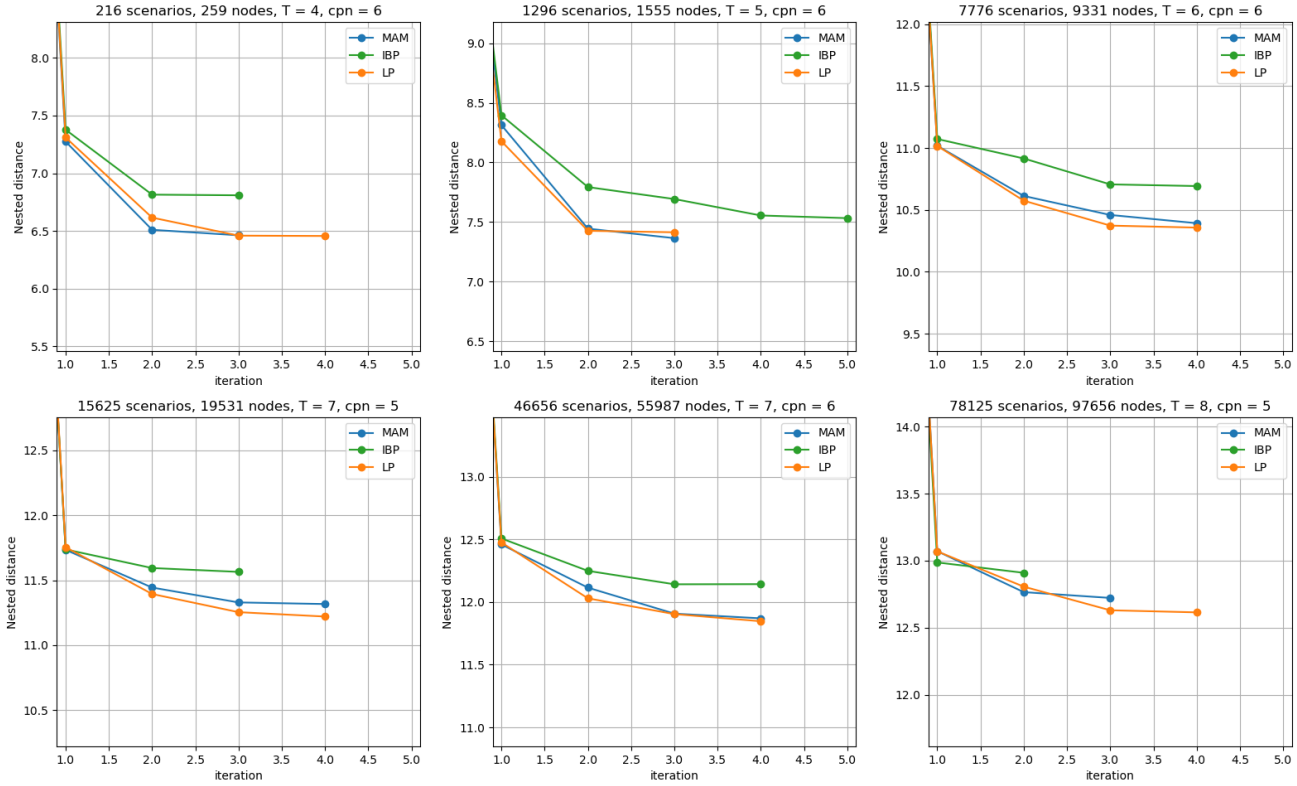


Figure 4.6: Evolution of the Nested Distance along the reduction iterations for different initial tree sizes with a zoom.

Table 4.1 shows that for small initial trees, up to 7776 scenarios, the LP variant is very efficient: it provides the lowest ND solution in the shortest time. But from 15625 scenarios and more, IBP is faster. As the number of scenarios increases, the relative performances of MAM get better and, in our case with more than 46656 scenarios, MAM is eventually twice faster than LP while reaching the same precision as depicted in Figure 4.5. As shown in the last graph, with even more nodes and scenarios (78125 and 97656, respectively), MAM is the fastest variant. Note that IBP is a very robust method: not only is the precision reached more than reasonable, according to Figure 4.5, but the total time of execution is always in the ballpark of the fastest execution time of all algorithms. Leveraging that MAM is parallelizable, we ran results using 4 processors. We witnessed that in this configuration, the variant MAM is the most advantageous one when the initial tree has more than 20000 scenarios by far.

4.4.2 Impact of the tree structure

We observed when using the different approaches to tackle real-life examples that the *structure* of the initial large tree has an impact on the convergence speed. Real-life trees do not span homogeneously from the root to the last stage, and when heterogeneity occurs, switching from one method to another can make the global reduction

Scenarios	Nodes	LP	IBP	MAM	MAM 4 processors
216	**259**	0.17	0.49	2.21	0.56
1296	**1555**	1.54	14.83	18.23	6.28
7776	**9331**	74.25	161.19	344.83	124.44
15625	**19531**	487.58	323.76	816.46	341.62
46656	**55987**	4905	2136	2541	1256
78125	**97656**	13797	4334	3458	1635

Table 4.1: Total time (in seconds) per method for the studied trees.

algorithm significantly faster.

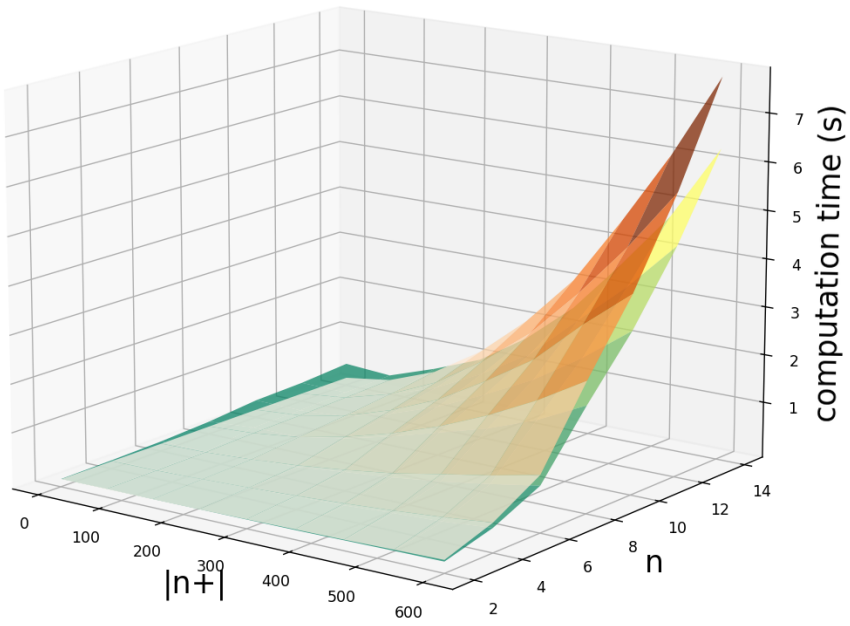


Figure 4.7: Influence of the tree structure on the computation time of a stage, depending on the method in use: MAM in green and LP in orange.

Figure 4.7 illustrates the tree structure influence on the computation time of a stage, depending on the algorithm used. The original large tree is built as follows: from the root, two nodes are spanning (stage $t = 2$), then stage $t = 3$ counts n nodes. From this configuration, we made the number $|n+|$ of children from these n nodes vary from 1 to 600 children (stage $t = T = 4$). To put it differently, we built a tree with n subtrees at *stage* 3 having dimension $|n+|$. We reduce this tree into a binary one and evaluate *stage* 3 reduction time[1]. We recall that the most expensive step of the algorithm consists in solving the greatest number of (WB) problems, thus stage 3.

For any number n of subtrees, the LP solver is always better when $|n+|$ is low, but when increasing $|n+|$ MAM solver ends up being faster because it treats better larger problems. This discrepancy gains momentum when the

[1]The computation time is evaluated as the mean of the five iterations along the reduction - at stage 3

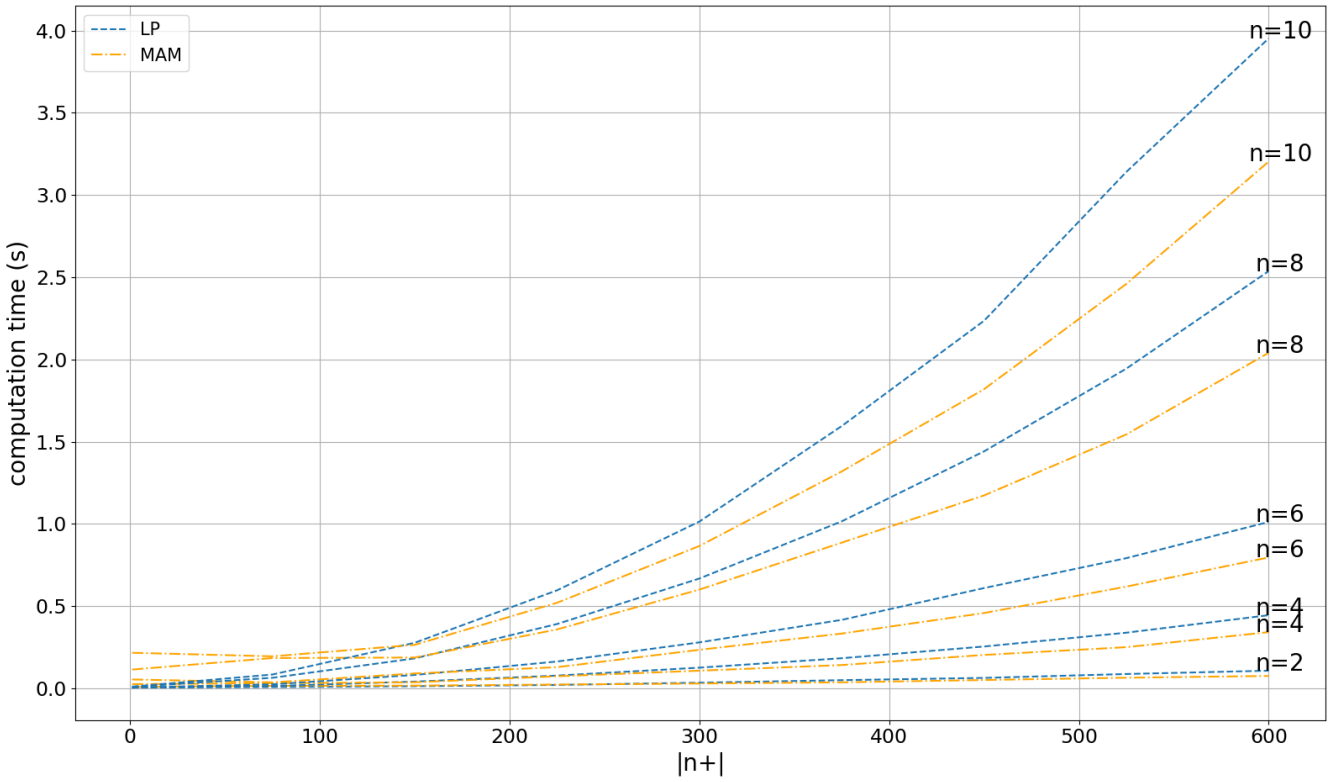
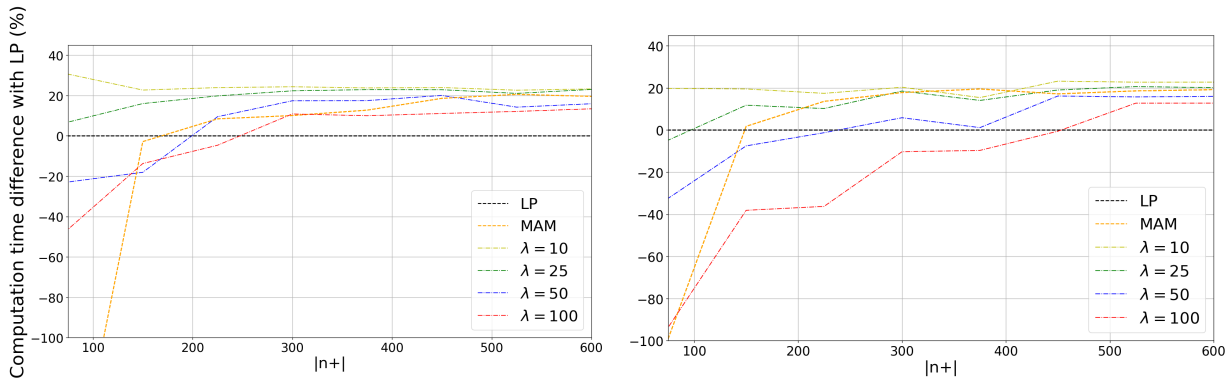


Figure 4.8: Influence of the tree structure on the computation time for small n.

number of subtrees rises. It is observed that the treshold is reached even faster when n is large. Therefore, at a stage with $n > 10$ (and $|n+| > 1$) where the reduction can take more than $4s$ we would advise to always use MAM. But, as illustrated in Figure 4.8, for smaller n we would have a closer look, and use MAM only if $|n+| > 150$, otherwise keep the LP method. Figure 4.7 and Figure 4.8 show that choosing the adequate method to tackle the reduction can speed up the computation time from 20% to 35%. In practice, it can be observed that this repartition ends up using half MAM (mostly for the deep stages where $t \gg 0$) and half LP (mostly for the stages close to the root and the heterogeneity in the structure) to reduce a real-life initial tree.

The IBP method is challenging to use in practice within the tree reduction algorithm because of the complexity involved in tuning the hyperparameter λ, which ideally needs to be carefully chosen for each barycenter computation to achieve acceptable precision. Fine-tuning is necessary to control its accuracy. If a broadly set hyperparameter λ is used, the algorithm may either encounter double-precision overflow errors at certain stages or compute a solution that significantly deviates from the exact optimization, without any guarantee or control over the resulting precision.

Figure 4.9 presents a study on the influence of a broad λ used for IBP in the tree reduction algorithm for the same datasets as earlier. It shows that the use of IBP enables the reduction tree algorithm to be fast for both small and large n and $n+$, and stresses that the smaller the λ the faster the reduction. But Figure 4.10 underlines that



(a) $n = 8$ (b) $n = 12$

Figure 4.9: Speed comparison with IBP for different λ: A positive time difference means the method is faster than LP. Each curve is obtained by averaging the ND accuracy over $n \in \{2, 4, 6, 8, 10, 12, 14, 16\}$.

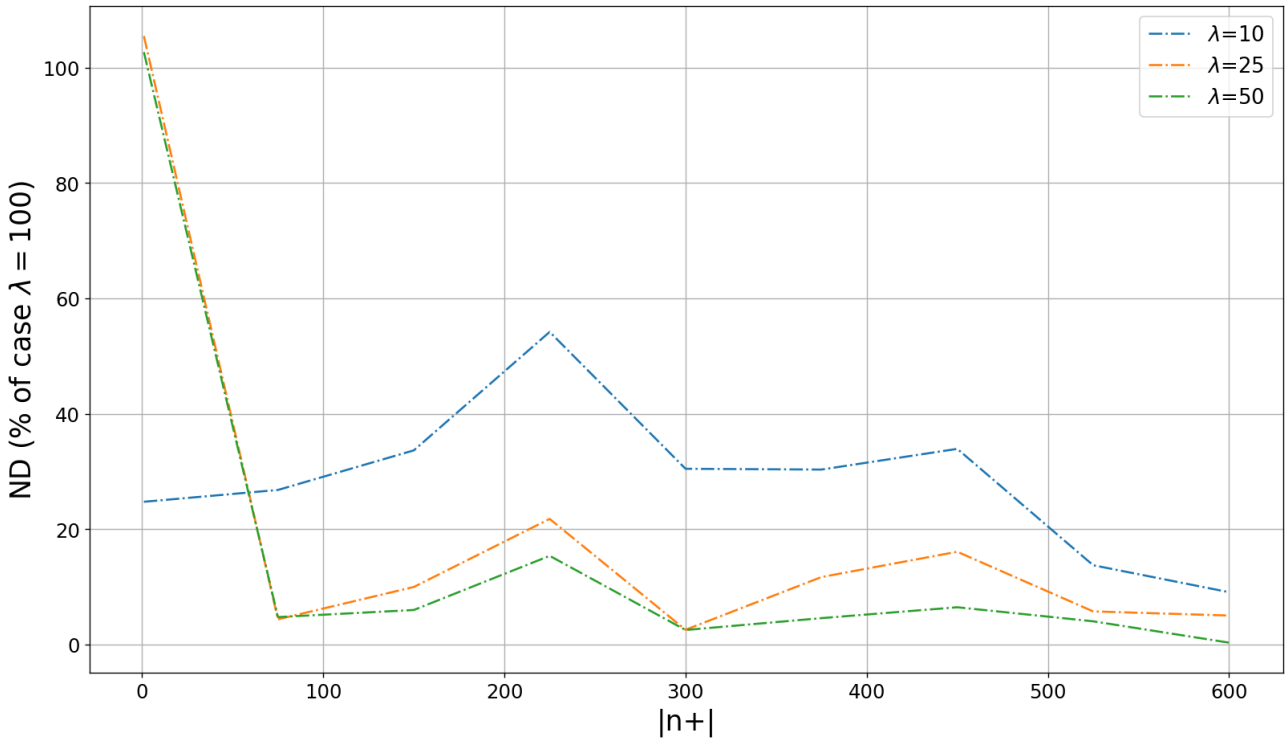


Figure 4.10: Average influence of λ in the precision. Each curve is obtained by averaging the ND accuracy over $n \in \{2, 4, 6, 8, 10, 12, 14, 16\}$.

this speed comes with a cost since the smaller the λ and the less accurate the computation of the barycenter - the

Scenario set	Filtrations	initial ND	reduced ND
1	**Kmeans**	2757	1219
	FFS	1384	658
2	**Kmeans**	1699	1092
	FFS	1936	896
3	**Kmeans**	1653	832
	FFS	1499	716
4	**Kmeans**	1858	963
	FFS	1161	566
5	**Kmeans**	1968	1054
	FFS	917	540

Table 4.2: Comparison of the ND to the original tree before and after tree reduction using different initialization techniques.

greater the ND. Figure 4.10 has been obtained by averaging the ND accuracy over n [2], where the case $\lambda = 100$ is taken as reference.

4.4.3 Impact of the initialization

In this section, we incorporate the reduction tree algorithm (utilizing either MAM or LP, depending on the results from Section 4.4.2) into a stochastic optimization pipeline. Data were collected from an industrial site in Solaize, France, where battery production and local consumption have been monitored over several years. Using this data, we generated scenarios for battery consumption and production throughout a single day, divided into 48 time steps, employing the methodology outlined in [4]. This approach allows us to create 100 $2D$ scenarios, each consisting of 48 stages with scenario values spanning from 0 to 25kW for the consumption and 0 to 40kW for the production. These scenarios can be modeled as a large tree. We reduce this tree using various initial filtrations to construct a smaller, approximate tree. To built these filtrations we leverage two scenario selection algorithms:

- *Kmeans method*, starting from 100 scenarios it creates 25 clusters using the Euclidean norm, and then computes the 25 corresponding barycenters;
- *Fast Forward Selection (FFS) method*, introduced by Heitsch and Römisch in [64]. The method iteratively selects scenarios that minimize the Wasserstein distance to the remaining scenarios. At each step, the scenario that best approximates the distribution is added to the reduced set until the desired number of 25 scenarios is reached, ensuring an efficient yet effective reduction.

From these reduced number of scenarios we generate the associated trees. They will initialize Algorithm 5. We make the experience multiple times by generating several sets of scenarios.

[2] $n \in \{2, 4, 6, 8, 10, 12, 14, 16\}$.

Table 4.2 compares the ND to the original large tree, both before and after tree reduction, highlighting the importance of the initial filtration. Note that the ND is consistently reduced after the tree reduction algorithm, as expected. When FFS is used to generate the initial filtration, the reduced approximated tree remains consistently closer to the original tree in terms of nested distance, even if the ND between the original tree and the FFS filtration is larger than that between the original tree and the K-means filtration at initialization. This is because FFS is a specialized algorithm designed to minimize the Wasserstein distance between the selected scenarios and the original ones, thereby preserving the maximum amount of information from the initial processes.

Chapter 5

Solving EMS problems

This chapter implements and compares different control strategies for real-world Energy Management System (EMS) problems, leveraging the previously developed methods to ensure fidelity to real-world conditions.

Abstract In this chapter, we consider a stationary battery using ground truth measurements of electricity consumption and production from a predominantly commercial building in France. This work is motivated by the fact that, on the one hand, classical approaches such as MPC, which deal with uncertainties by computing deterministic solutions in a receding horizon fashion, can lead to suboptimal economic performance; on the other hand, risk-free multi-stage stochastic programming lacks robustness when reality deviates from forecast. To achieve a good trade-off between robustness and performance, we explore and compare other control models including robust stochastic optimization models and Reinforcement Learning strategies. Through numerical experiments, we evaluate these models in terms of cost efficiency, computational scalability, and out-of-sample robustness, offering a comprehensive comparison and insights into their practical interest for real-world EMS problems.

5.1 Litterature review

Energy Management Systems (EMS) play a central role in optimizing electricity consumption, and reducing operational costs in modern power networks. However, managing energy efficiently requires making sequential decisions under uncertainty: the future is not fully known when decisions must be made. Traditionally, such problems are addressed using stochastic programming models, which aim to minimize expected costs by simulating a wide range of future scenarios. These models assume that the probability distribution of future data is known in advance. Although reasonable, this assumption is not always satisfied in practice, and the true distribution sometimes has to be estimated from limited historical data, which may affect the reliability of the results derived from it.

To overcome this limitation, practitioners often rely on Model Predictive Control (MPC) [58,71,88], a control model that solves a deterministic optimization problem at each stage using the most recent data and a single scenario for

predicting the unknown variables. While MPC is simple to implement and widely adopted in industry, it does not explicitly account for future uncertainties and may result in suboptimal performance [55].

Another strategy is Robust Optimization (RO) [55, 116], which discards the need for probabilities altogether. Instead, it assumes the worst-case scenario among all the possible outcomes. This approach yields conservative solutions, which are often too costly or overly cautious for practical use [59].

A more balanced approach, known as Distributionally Robust Optimization (DRO), considers distributions lying within an ambiguity set – a set of plausible probability distributions constructed from a finite collection of scenarios. The goal is to optimize decisions to perform well against the worst-case distribution in this set. In fact, rather than relying on a single estimated probability measure, DRO acknowledges the uncertainty in this estimation and instead considers all distributions that lie within a certain neighborhood of the empirical measure. The quality of the DRO solution strongly depends on how this ambiguity set is defined as it captures the trade-off between robustness and reliance on a specific distribution. Among distributionally robust approaches, recent works have focused on the use of Wasserstein distance [51, 72] to define ambiguity sets. While the Wasserstein distance offers attractive mathematical properties, allowing for models to be both robust, its application to multistage problems raises nontrivial challenges [44, 110].

The recent work [80], situated within the field of risk-averse stochastic optimization, introduces a novel approach that integrates variance penalization directly into multistage SP models. By accounting for cost variability across scenarios, the approach effectively balances robustness and performance.

A key challenge shared by most of the optimization-based approaches discussed so far is their reliance on scenario generation. While increasing the number of scenarios can enhance solution accuracy, it also leads to substantial computational overhead. Although scenario selection and tree reduction techniques offer partial remedies [12, 64, 70, 86], they do not eliminate the fundamental dependency on pre-specified scenario sets.

An alternative to optimization-based models is Reinforcement Learning (RL), which learns decision policies directly from data through interaction with the environment [24, 32, 127]. RL avoids the need to specify probability distributions or generate accurate scenarios. Instead, it learns from repeated experience to make sequential decisions under uncertainty. Although promising, RL often requires a large amount of data and may struggle with constraints typically present in EMS problems [73].

This chapter systematically compares several approaches, including MPC, (risk-free and risk-averse) stochastic programming models, robust optimization, distributionally robust models, and reinforcement learning for a specific EMS application. We evaluate each approach in terms of robustness, computational efficiency, and out-of-sample performance, providing guidance on their practical use in real-world energy systems.

The remainder of this chapter is structured as follows. Section 5.2 provides the notations that will be used in this chapter. Section 5.3 introduces the EMS problem we study and provides the mathematical background for the models discussed previously. Section 5.4 presents the computational methods, along with the specific algorithmic parametrizations tailored to address the EMS problem. Finally, Section 5.5 reports numerical results, comparing the different approaches after performing cross-validation to ensure each method is tuned for the best performance on the EMS task.

5.2 Notation

Let X be a Hilbert space, we denote by $\langle .,. \rangle_{\mathrm{X}}$ its inner product, and by $\|.\|_X$ its corresponding norm.

Let E $\subset$ X be convex, we denote $\mathrm{i}_{\mathrm{E}} : \mathrm{X} \mapsto \mathbb{R} \cup \{+\infty\}$ the indicator function of E, i.e. $\mathrm{i}_E(x) = 0$ if $x \in E$ and $\mathrm{i}_E(x) = +\infty$ otherwise.

Additionally we denote $\mathbb{1}_E : X \mapsto \{0, 1\}$ the characteristic function of E, i.e. $\mathbb{1}_E(x) = 1$ if $x \in E$ and $\mathbb{1}_E(x) = 0$ otherwise .

Given a Fréchet-differentiable function $f : \mathrm{X} \mapsto \mathbb{R}$ we denote $f' \in \mathrm{X}$ the Fréchet-derivative of f.

Let $(\Omega, \mathcal{F}, P)$ be a probability space, we denote random variables from Ω to X using bold characters such as $\boldsymbol{\xi} : \Omega \mapsto \mathrm{X}$. If Ω is a discrete set, i.e. $\Omega = \{\omega_1, \dots, \omega_S\}$ we denote indifferently $\boldsymbol{\xi}(s)$ or ξ^s the outcome $\boldsymbol{\xi}(\omega_s)$ of the random variable.

We denote with blackboard capital letters sets of random variable such as $\mathbb{X} := \{\boldsymbol{x} : \Omega \mapsto \mathrm{X}\}$. We denote $\mathbb{E}_P$ the mathematical expectation with respect to the probability measure P.

The random Hilbert space $\mathbb{X}$ is endowed with the inner product $\langle .,. \rangle_{\mathbb{X}} := \mathbb{E}\left(\langle .,. \rangle_{\mathrm{X}}\right)$, when this quantity is finite, and the corresponding norm $\|.\|_{\mathbb{X}} := \mathbb{E}(\|.\|_X)$.

Given $p \in [1, +\infty]$, we denote $\mathrm{L}^p(A; B)$ (or L^p) the Lebesgue spaces of functions from A to B and we denote $\|.\|_{\mathrm{L}^p}$ the corresponding p-norm.

For all $1 \leq p < +\infty$, we denote $\mathbb{L}^p$ the space of random variables $\boldsymbol{\xi} : \Omega \mapsto \mathrm{L}^p$ and we denote $\|\boldsymbol{\xi}\|_{\mathbb{L}^p} := \mathbb{E}(\|\boldsymbol{\xi}\|_{\mathrm{L}^p}^p)^{\frac{1}{p}}$.

We denote $\mathbb{L}^\infty$ the space of random variables $\boldsymbol{\xi} : \Omega \mapsto \mathrm{L}^\infty$ and we denote $\|\boldsymbol{\xi}\|_{\mathbb{L}^\infty} := \inf\{y \in \mathbb{R} : \mu(\{\omega \in \Omega : \|\boldsymbol{\xi}(\omega)\|_{\mathrm{L}^\infty} > y\}) = 0\}$.

Let X be a set, and let $x \in X$, we denote δ_x the Dirac delta function on x. Finally, let $x \in \mathbb{R}^n$, we denote x_k the k^{th} coordinate of x and let $M \in \mathbb{R}^{n \times m}$, we denote $M_{k,l}$ the value of the k^{th} row and l^{th} column of M.

5.3 Problem statement

We consider a Stochastic Optimal Control Problem (SOCP) modeling the operation of a stationary battery connected downstream of a prosumer's electricity meter. A *prosumer* is an end-user which both consumes and produces electricity, typically via intermittent sources such as photovoltaic panels, as depicted in Figure 5.1. Often the goal is to minimize the expected electricity cost over a finite time horizon, while accounting for battery dynamics and the stochastic nature of both **consumption** and **production**.

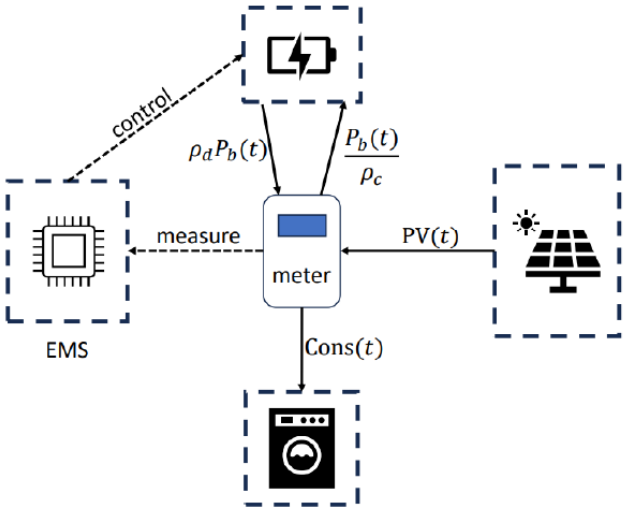


Figure 5.1: Schematic diagram of a domestic system with a stationary battery controlled by an EMS [80]

The EMS battery strategy Let P_b, **Cons**, and **PV** be respectively the battery's charging power, the electric consumption and photovoltaic power both measured at the meter. The electricity bill to minimize, as a function of these variables, writes

$$J_{t_1:t_2}(P_b, \mathbf{Cons}, \mathbf{PV}) := \int_{t_1}^{t_2} p_r^c(t) \max\{\boldsymbol{P_m}(t), 0\} + p_r^d(t) \min\{\boldsymbol{P_m}(t), 0\} \mathrm{d}t \tag{5.1}$$

where $p_r^c(t)$ [€/kWh] (resp. $p_r^d(t)$) is the buying (resp. selling) prices of electricity at time t. They satisfy $0 \leq p_r^d(t) \leq p_r^c(t)$ at all times, therefore eq. (5.1) is convex. The power measured at the meter, $\boldsymbol{P_m}$ [kW] writes

$$\boldsymbol{P_m}(t) := \mathbf{Cons}(t) - \mathbf{PV}(t) + \frac{1}{\rho_c} \max\{P_b(t), 0\} + \rho_d \min\{P_b(t), 0\}, \tag{5.2}$$

The parameters $\rho_c = 0.97$, and $\rho_d = 0.97$ represent the battery charge and discharge efficiencies, respectively. Now, the battery's dynamics are governed by

$$\dot{E}(t) = P_b(t), \tag{5.3}$$

and the operational constraints of the battery's charging / discharging process are

$$E \in \mathrm{L}^\infty([t_1, t_2]; [0, 13 \text{ kWh}]), \tag{5.4a}$$

$$P_b \in \mathrm{L}^2([t_1, t_2]; [-8, 8\rho_c \text{ kW}]), \tag{5.4b}$$

$$E(t_1) = E(t_2) = E^0. \tag{5.4c}$$

In this formulation, $E(t)$ [kWh] represents the battery's State of Energy (SoE) at time t, while E^0 denotes the initial and terminal state of energy, ensuring periodicity over the time horizon. The interactions between the sources of power are depicted in the diagram of Figure 5.1. For the sake of readability, we denote C the set of admissible deterministic charging powers defined as follows

$$\mathrm{C} := \left\{ P_b \in \mathrm{L}^2 : \text{ eqs. (5.3) and (5.4) hold} \right\}. \tag{5.5}$$

5.4 Computing solutions for different strategies

In this work, we are interested in exploring different optimization models for our EMS problem taking in account uncertainties on consumption and (photovoltaic) production. We start by describing a standard model predictive control formulation.

5.4.1 Model predictive control

At each time step t, MPC algorithms [58,71,88] consist of predicting a realization of the random variables **Cons** and **PV**, respectively denoted $\widehat{\text{Cons}}$ and $\widehat{\text{PV}}$, on a defined and finite horizon ΔT, and solve the following deterministic optimization problem:

$$\min_{P_b \in \text{C}} J_{t:t+\Delta T}(P_b, \widehat{\text{Cons}}, \widehat{\text{PV}}), \tag{MPC}$$

and then apply $P_b(t)$. Note that, in the MPC framework, the charging battery power (P_b) and the energy in the battery (E), are not random variables since they only depend on the predicted scenario $\widehat{\text{Cons}}$, $\widehat{\text{PV}}$ which is deterministic. This MPC strategy is a rolling horizon method of actualization horizon $h < \Delta T$. This method is easy to implement and relies on deterministic optimization algorithms. However, the solutions can be far from optimal depending on the accuracy of the single predicted scenario ($\widehat{\text{Cons}}$, $\widehat{\text{PV}}$).

5.4.2 Scenario-based methods

5.4.2.1 Mathematical framework

The control methods presented in this section all rely on a finite set of scenarios drawn from a given probability distribution. In the following we denote $\Xi_{t_1:t_2}$ the discrete set of scenarios defined as follows

$$\Xi^S_{t_1:t_2} := \left\{ \text{Cons}^s - \text{PV}^s \in \text{L}^2([t_1, t_2]; \mathbb{R}) : s = 1, \dots, S \right\}. \tag{5.6}$$

With each scenario, $\text{Cons}^s - \text{PV}^s$, is associated with a probability $p_s > 0$, satisfying $\sum_{s=1}^S p_s = 1$. For notation simplicity, we denote $\xi^s := \text{Cons}^s - \text{PV}^s$. Scenario based methods consist in computing an optimal control for each scenario $\xi^s \in \Xi_{t_1:t_2}$, that is to say, S charging/discharging chronicles, denoted P_b^s. However, these S optimal controls are not independent as they must satisfy a Δ-non-anticipative constraint: If $s \neq s'$ are such that $\xi^s(t) = \xi^{s'}(t)$, for all $t \leq \tau$, then necessarily $P_b^s(t) = P_b^{s'}(t)$, for all $t \leq \tau + \Delta$. This Δ-non-anticipativity constraint allows for taking into account the measurement delays. Indeed, consumption at time t, Cons(t) is the mean power consumption on the time interval $[t, t + \Delta]$ and therefore is only measured at time $t + \Delta$. Thus, the battery power P_b on $[t, t + \Delta]$ must be computed without knowledge of the consumption nor the PV production on this time interval hence the Δ-non-anticipativity constraint. Throughout the chapter, we denote $\mathcal{N}_\Delta$ the set of Δ-non-anticipative controls. Mathematical details on scenario-based methods can be found in [28,103,111].

Therefore, the corresponding set of admissible random charging powers, associated with set C and accounting for non-anticipativity, is

$$\mathbb{C} := \{ \boldsymbol{P_b} \in \mathcal{N}_\Delta : \boldsymbol{P_b} \in \text{C almost surely} \}. \tag{5.7}$$

5.4.2.2 Deterministic-control approach for SP

The deterministic-control approach to stochastic programming (SP) models involves computing a single control policy that minimizes the expected electricity bill across all scenarios. This contrasts with standard SP models, which seek a scenario-dependent, non-anticipative control P_b^s, for $s = 1, \dots, S$. The deterministic control approach is workable due to the problem's structure: the set of solutions does not depend on uncertainty.

Definition 9 (Deterministic-Control Approach for SP (DSP))**.** *The DSP problem is defined as*

$$\inf_{P_b \in \mathrm{C}} \left[\sum_{s=1}^{S} p_s J_{t_1:t_2}(P_b, \mathrm{Cons}^s, \mathrm{PV}^s) \right], \tag{DSP}$$

where $\mathrm{Cons}^s - \mathrm{PV}^s \in \Xi_{t_1:t_2}^S$ is the s-th scenario.

Definition 9 fits a standard setting of constrained optimal control problem and can be solved using numerical methods such as [79]. This model's main advantage is that it remains numerically tractable even when using a large number of scenarios. This method is highly robust but can be sub-optimal.

5.4.2.3 Stochastic programming with variance penalization

The stochastic programming with variance penalization consists of solving the following optimal control problem, parameterized with the parameter $\alpha \geq 0$

Definition 10 (SP with variance penalization)**.** *The SP model with variance penalization consists in solving for* $\boldsymbol{P_b}$ *the following optimal control problem*

$$\inf_{\boldsymbol{P_b} \in \mathbb{C}} \left[\sum_{s=1}^{S} p_s J_{t_1:t_2}(\boldsymbol{P_b}(s), \mathrm{Cons}^s, \mathrm{PV}^s) + \frac{\alpha}{2} \sum_{s=1}^{S} p_s \|\boldsymbol{P_b}(s) - \sum_{s'=1}^{S} p_{s'} \boldsymbol{P_b}(s')\|_{\mathrm{L}^2}^2 \right], \tag{VSP}$$

where $\boldsymbol{P_b}$ is the discrete random variable associating with each scenario s the optimal charging power $\boldsymbol{P_b}(s) := P_b^s$, and where $\mathrm{Cons}^s - \mathrm{PV}^s \in \Xi_{t_1:t_2}^S$.

For $\alpha = 0$, Definition 10 reduces to the standard risk-neutral optimal control problem [103]. For $\alpha = +\infty$, the problem consists of finding a deterministic battery's charging/discharging power that minimizes the electricity bill's expectation, i.e., (VSP) boils down to (DSP). For $\alpha \in (0, +\infty)$, Definition 10 balances the risk-neutral and Definition 9. The solving algorithm, presented in [80], for Definition 10 is displayed in Algorithm 6 below. We highlight that Step 1 of this algorithm is the only computationally intensive step and consists of solving S independent deterministic optimal control problems. Efficient algorithms such as [26, 79] are available for this task.

Let $\Lambda(s,t) := \left\{ j \in \{1, \dots, S\} \mid \xi_l^j = \xi_l^s, \forall l \leq t - \Delta \right\}$ be the set of scenarios sharing the same history as scenario s up to stage t.

Algorithm 6 Regularized Progressive Hedging Algorithm (RPHA)

1: **Initialization:** Set $\boldsymbol{z}^0 \in \mathbb{L}^2$, $\boldsymbol{\lambda}^0 \in \mathcal{N}_\Delta^\perp$, $r > 0$, tolerance $\texttt{Tol} > 0$, $k \leftarrow 0$ and success $\leftarrow False$
2: **while** not success **do**

▷ Step 1: Solve for each scenarios indexed by s

3:

$$\boldsymbol{P_b}^{k+1}(s) \;\leftarrow\; \arg\min_{P_b \in \mathrm{C}} \left\{ J_{t_1:t_2}(P_b, \mathrm{Cons}^s, \mathrm{PV}^s) \;+\; \langle \boldsymbol{\lambda}^k(s), P_b \rangle_{\mathrm{L}^2} \;+\; \frac{r}{2} \| P_b \;-\; \boldsymbol{z}^k(s) \|_{\mathrm{L}^2}^2, \quad s \;=\; 1, \ldots, S \right\}$$

▷ Step 2: Project onto the orthogonal non-anticipative space

4: $\boldsymbol{\lambda}^{k+1} \leftarrow \boldsymbol{\lambda}^k + r\mathrm{Proj}_{\mathcal{N}_\Delta^\perp}(\boldsymbol{P_b}^{k+1})$

▷ Step 3: Update the dual variables

5: $\boldsymbol{z}^{k+1} \leftarrow \boldsymbol{z}^k - \boldsymbol{P_b}^{k+1} + \frac{\alpha}{r+\alpha} \sum_s p_s \left(2\boldsymbol{P_b}^{k+1}(s) - \boldsymbol{z}^k(s) \right) + \frac{r}{r+\alpha} \mathrm{Proj}_{\mathcal{N}_\Delta} \left(2\boldsymbol{P_b}^{k+1} - \boldsymbol{z}^k \right)$

▷ Step 4: Check convergence

6: success $\leftarrow \| \boldsymbol{P_b}^{k+1} - \boldsymbol{P_b}^k \|_{\mathbb{L}^2} < \texttt{Tol}$
7: **end while**

Then, for $\boldsymbol{z} \in \mathbb{L}^2$, the projection $\boldsymbol{x} = \mathrm{Proj}_{\mathcal{N}_\Delta}(\boldsymbol{z})$ is given component-wise by

$$\boldsymbol{x}_t(s) = \frac{1}{\sum_{j \in \Lambda(s,t)} p_j} \sum_{j \in \Lambda(s,t)} p_j \boldsymbol{z}_t(j), \quad t = 1, \ldots, T, s = 1, \ldots, S. \tag{5.8}$$

In words, for each stage t and scenario s, $\boldsymbol{z}_t(s)$ is replaced by the average over all scenarios that share the same history up to t. The projection onto the orthogonal set is

$$\mathrm{Proj}_{\mathcal{N}_\Delta^\perp}(\boldsymbol{z}) = \boldsymbol{z} - \mathrm{Proj}_{\mathcal{N}_\Delta}(\boldsymbol{z}). \tag{5.9}$$

5.4.2.4 Distributionally Robust Optimization (DRO)

In this setting, we use two fixed sets of scenarios, denoted Ξ^S and Ξ^L of respective size S and L with $L \geq S$. Each scenario $\xi^l := \mathrm{Cons}^l - \mathrm{PV}^l \in \Xi^L$ is associated with a probability p_l. The DRO setting consists of solving the following problem:

Definition 11 (Distributionnaly Robust Problem)**.**

$$\inf_{\boldsymbol{P_b} \in \mathbb{C}} \sup_{q \in \mathcal{P}_\theta} \left[\sum_{s=1}^{S} q_s J(\boldsymbol{P_b}(s), \mathrm{Cons}^s, \mathrm{PV}^s) \right], \tag{DRO}$$

where $\mathcal{P}_\theta$ is the ambiguity set defined as follows

$$\mathcal{P}_\theta := \left\{ q \in \mathbb{R}_+^S : \sum_s q_s = 1, \mathrm{W}_2 \left(\sum_{s=1}^{S} q_s \delta_{\xi^s}, \sum_{l=1}^{L} p_l \delta_{\xi^l} \right) \leq \theta \right\}. \tag{5.10}$$

Here W_2 is the 2-Wasserstein distance between two discrete probability measures [106, Ch. 5] and Chapter 2. The parameter θ controls the size of the ambiguity set. The larger θ is, the larger the ambiguity set becomes. Thus, large values of θ lead to a worst-case type behavior by assigning all probabilities q_s except one to zero, which is equivalent to selecting the worst scenario over Ξ^S as in RO. Conversely, when θ is small, the ambiguity set is also small, and the weights q_s are adjusted so that the probability defined over Ξ^S is close to that defined over Ξ^L, approaching a risk-neutral problem.
Before describing an algorithm to solve (DRO), we need to introduce two mathematical operations, the projection onto the ambiguity set $\mathcal{P}_\theta$ from eq. (5.10) and the projection onto the epigraph of the optimal control function described in eqs. (5.1), (5.3) and (5.4).

Definition 12 (Projection onto the ambiguity set $\mathcal{P}_\theta$)**.** *Let $p \in \mathbb{R}^S$, let $\boldsymbol{\xi} \in \Xi^S$ and $\hat{\boldsymbol{\xi}} \in \Xi^L$, we denote the projection of p onto $\mathcal{P}_\theta$*

$$\mathrm{Proj}_{\mathcal{P}_\theta}(p) := \bar{x} \tag{5.11}$$

where $\bar{x}$ is defined as follows

$$(\bar{x}, \bar{\eta}) \in \underset{x \in \mathbb{R}^S_+, \eta \in \mathbb{R}^{S \times L}_+}{\arg\min} \|x - p\|^2 \tag{5.12}$$

such that

$$\sum_{s=1}^{S} \sum_{l=1}^{L} \eta_{s,l} \|\xi^s - \hat{\xi}^l\|^2_{\mathrm{L}^2} \leq \theta \tag{5.13a}$$

$$\sum_{s=1}^{S} \eta_{s,l} = p_l, \quad l = 1, \dots, L \tag{5.13b}$$

$$\sum_{l=1}^{L} \eta_{s,l} = x_s, \quad s = 1, \dots, S \tag{5.13c}$$

$$\sum_{s=1}^{S} \sum_{l=1}^{L} \eta_{s,l} = 1. \tag{5.13d}$$

Thus, the projection onto the ambiguity set is a quadratic problem of size $S(1 + L)$, therefore, computationally intensive depending on S and L.

Definition 13 (Projection onto the epigraph)**.** *First, let us consider the following deterministic optimal control problem*

$$\bar{J}^s_{t_1:t_2}(P_b) := J_{t_1:t_2}(P_b, \mathrm{Cons}^s, \mathrm{PV}^s) + \mathrm{i}_{\mathrm{C}}(P_b)$$

where C *is defined in eq. (5.5). Let $(P_b, \rho) \in \mathrm{L}^2 \times \mathbb{R}$, we denote $\mathrm{Proj}_{\mathrm{epi}\, \bar{J}^s_{t_1:t_2}}(P_b, \rho)$ the corresponding projection onto the epigraph of $\bar{J}^s_{t_1:t_2}$ defined as the solution of the following optimal control problem*

$$\mathrm{Proj}_{\mathrm{epi}\, \bar{J}^s_{t_1:t_2}}(P_b, \rho) \in \underset{P \in \mathrm{L}^2, \lambda \in \mathbb{R}}{\arg\min} \|P - P_b\|^2_{\mathrm{L}^2} + (\rho - \lambda)^2 \tag{5.14}$$

under the following constraints

$$\dot{E}(t) = P(t) \tag{5.15a}$$
$$\dot{Q}(t) = p_r^c(t)\max\{P_m(t),0\} + p_r^d(t)\min\{P_m(t),0\} \tag{5.15b}$$
$$P_m(t) = \mathrm{Cons}^s(t) - \mathrm{PV}^s(t) + \frac{1}{\rho_c}\max\{P(t),0\} + \rho_d \min\{P(t),0\} \tag{5.15c}$$
$$Q(t_1) = 0 \tag{5.15d}$$
$$Q(t_2) - \lambda \leq 0 \tag{5.15e}$$
$$E(t_1) = E(t_2) = E^0. \tag{5.15f}$$

This is a standard deterministic optimal control problem and can be efficiently solved numerically.

We are now ready to detail the solving algorithm for problem (DRO) which is a direct adaptation of [44] to the problem at hand. The second step of Algorithm 7, i.e. the projection onto the ambiguity set, is the bottleneck of the algorithm. Indeed, the complexity of this task dramatically increases with the sizes of the scenarios sets. On the other hand, the projection onto the epigraphs performed at Step 3 is parallelizable with respect to the scenarios, and each individual task is a standard, easy-to-solve optimal control problem.

Algorithm 7 Scenario Decomposition with Alternating Projections - SDAP [44]

Initialization: Set $\boldsymbol{z}_{P_b}^0 \in \mathbb{L}^2$, $z_u^0 \in \mathbb{R}^S$, $z_v^0 \in \mathbb{R}^S$, $r > 0$, two tolerances $\mathrm{Tol}_{P_b}, \mathrm{Tol}_J > 0$, set $k \leftarrow 0$, and converged $\leftarrow False$
while not converged **do**
▷ Step 1: Projection onto $\mathcal{N}_\Delta$
  Define $\boldsymbol{P_b}^k \leftarrow \mathrm{Proj}_{\mathcal{N}_\Delta}(\boldsymbol{z}_{P_b}^k)$, $u^k \leftarrow \frac{z_u^k + z_v^k}{2}$
▷ Step 2: Projection on $\mathcal{P}_\theta$:
  $\hat{v}^{k+1} \leftarrow (2u^k - z_v^k) - \frac{1}{r}\mathrm{Proj}_{\mathcal{P}_\theta}\left(r(2u^k - z_v^k)\right)$
▷ Step 3: Projection onto the epigraphs
  for $s = 1, \ldots, S$ **do**
    $(\hat{\boldsymbol{P}}_b^{k+1}(s), \hat{u}_s^{k+1}) \leftarrow \mathrm{Proj}_{\mathrm{epi}\,\bar{J}_{t_1:t_2}^s}\left(2\boldsymbol{P_b}^k(s) - \boldsymbol{z}_{P_b}^k(s), 2u_s^k - z_{u\,s}^k\right)$
  end for
▷ Step 4:
  $\boldsymbol{z}_{P_b} \leftarrow \boldsymbol{z}_x^k + \hat{\boldsymbol{P}}_b^{k+1} - \boldsymbol{P_b}^k$
  $z_u^{k+1} \leftarrow z_u^k + \hat{u}^{k+1} - u^k$
  $z_v^{k+1} \leftarrow z_v^k + \hat{v}^{k+1} - u^k$
▷ Step 5: Check Convergence
  converged $\leftarrow \|\hat{\boldsymbol{P}}_b^{k+1} - \boldsymbol{P_b}^k\| < \mathrm{Tol}_{P_b}$ and $\|\hat{u}^{k+1} - u^k\| \leq \mathrm{Tol}_J$ and $\|\hat{v}^{k+1} - u^k\| \leq \mathrm{Tol}_J$ $k \leftarrow k+1$
end while

5.4.3 Reinforcement learning

In reinforcement learning (RL), an agent learns an optimal behavior by interacting with an environment and receiving costs (or rewards) from these interactions. To solve the EMS problem, we will use a tabular Q-learning

algorithm inspired by [127]. We define a finite horizon Markov Decision Process (MDP) as $(\mathcal{T}, \mathcal{S}, \mathcal{A}, \mathbb{P}, c)$, where $\mathcal{T} = \{t_1, t_1+\Delta, \dots, t_2\}$ represents the finite time horizon, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$, $\mathbb{P}(s'|s,a)$ is the transition probability of passing from state s to state s' given action a, and c is the cost function representing the cost received after taking action a in state s. We choose the state to be $s := (E, \bar{\xi}) \in \mathcal{S}$, where:

- $E \in \{0, 0.13, \cdots, 13 \text{ kWh}\}$ is the energy state of the battery and, where the discretization step of 0.13 kWh is taken from [127];
- $\bar{\xi} = \text{Cons} - \text{PV} \in \{-8\times 2, -16+1, \cdots, 16\rho_c \text{ kW}\}$ is the difference between the electricity demand and production, where the discretization step of 1 kW is taken from [127].

In the following we introduce the notation $s^\tau := (E(\tau), \bar{\xi}(\tau))$ to denote the state at time τ. We set the action to be the battery charging power and thus the action space $\mathcal{A}$ regroups the feasible battery powers $\mathcal{A} := \{-8, 0, 8\rho_c \text{ kW}\}$, where the discretization this choice for $\mathcal{A}$ is motivated by [127].
Finally, $c^\tau : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is the cost function at time τ:

$$c^\tau(s, P_b) = p_r^c(\tau)\max\{P_m^\tau, 0\} + p_r^d(\tau)\min\{P_m^\tau, 0\}, \tag{5.16}$$

with

$$P_m^\tau = \bar{\xi}^\tau + \frac{1}{\rho_c}\max\{P_b, 0\} + \rho_d \min\{P_b, 0\}. \tag{5.17}$$

In this study, we consider deterministic policies: in state s at time τ, we define π so that the action taken at time τ is entirely defined by $P_b^\tau = \pi(\tau, s) \in \mathcal{A}$. The goal is to find a policy π^* minimizing the expectation of its finite horizon return: $\mathbb{E}_\pi\left[\sum_{\tau=t}^{t_2} c^\tau(\boldsymbol{s}^\tau, \boldsymbol{P}_b^\tau)\right]$.

The Q-function associated with a policy π is defined as:

$$Q^\pi(t, s, P_b) = \mathbb{E}\left[\sum_{\tau=t}^{t_2} c^\tau(\boldsymbol{s}^\tau, \boldsymbol{P}_b^\tau) \mid \boldsymbol{s}^t = s, \boldsymbol{P}_b^t = P_b\right]. \tag{5.18}$$

It is the expected value of choosing action P_b in state s starting at time step t and following policy π afterwards. The optimal Q-function Q^* is defined as $\forall(t, s, P_b) \in \mathcal{T}\times\mathcal{S}\times\mathcal{A}$, $Q^*(t, s, P_b) = \min_\pi Q^\pi(t, s, P_b)$, and it satisfies the Bellman optimality equation:

$$Q^*(t, s, P_b) = \mathbb{E}_{s'}\left[c^t(s, P_b) + \min_{\bar{P}_b \in \mathcal{A}} Q^*(t+\Delta t, s', \bar{P}_b)\right], \tag{5.19}$$

with $s' \sim \mathbb{P}(\cdot|s, P_b)$. Then the optimal policy is simply $\pi^*(t, s) = \arg\min_{P_b\in\mathcal{A}} Q^*(t, s, P_b)$.

The controlled process $\boldsymbol{E}$ satisfies (5.3), hence we can avoid the exploration of the controlled part of the state space by parallel simulations for every pairs of $(E, P_b) \in \{0, dE, \cdots, 13 \text{ kWh}\}\times\mathcal{A}$. The exploration will be needed only for the unknown process $\bar{\boldsymbol{\xi}}$. Therefore, instead of standard Q-learning [125], our RL agent will learn from a dataset $\mathcal{D}$ consisting of previously collected transitions on the difference between consumption and production: $\mathcal{D}$ contains $|\mathcal{D}|$ days of historical data of **Cons** and **PV**. We denote $N(s)$ the number of times s is present in the dataset $\mathcal{D}$.

Algorithm 8 Offline Q-Learning (Training)

Input: Given a training dataset $\mathcal{D}$, initialize $N(s) = 0$ for all $(t, s) \in \mathcal{T} \times \mathcal{S}$.
Initialize $Q(t, s, P_b) = 0$ for all $(t, s, P_b) \in \mathcal{T} \times \mathcal{S} \times \mathcal{A}$
for $t = t_2, t_2 - \Delta, \cdots, t_1$ **do**
  for each sample $((t, \bar{\xi}), (t + \Delta, \bar{\xi}')) \in \mathcal{D}$ **do**
    For all (E, P_b), collect state-action pairs $(s, P_b) = (E(t), \bar{\xi}(t), P_b)$ to be updated, and update the visit times of these state $N(s) \leftarrow N(s) + 1$
    Compute temporal difference error for all collected state-action pairs: $\mathrm{TD}(s, P_b) \leftarrow c(s, P_b) + \min_{\tilde{P}_b} Q(t + \Delta, s', \tilde{P}_b) - Q(t, s, P_b)$
    Update Q-function for all collected state-action pairs: $Q(t, s, P_b) \leftarrow Q(t, s, P_b) + \frac{1}{N(s)} \cdot \mathrm{TD}(s, P_b)$
  end for
end for
Return: Learned policy $\pi(t, s) = \arg\min_{P_b \in \mathcal{A}} Q(t, s, P_b)$

Given a dataset $\mathcal{D}$, the detailed training process is given in Algorithm 8. Then a policy π is derived as

$$\pi(t, s) = \arg\min_{P_b \in \mathcal{A}} Q(t, s, P_b), \quad \forall (t, s) \in \mathcal{T} \times \mathcal{S}. \tag{5.20}$$

The obtained policy π will be applied like a feedback controller on new days: at each timestep t, the state E and $\bar{\xi}$ over the last intervale $[t - \Delta, t]$) is observed, then an action P_b is chosen according to the policy π, and the action $P_b = \pi(t, s)$ is applied to the system – see Algorithm 9.

Algorithm 9 Offline Q-learning (Application on new days)

Given a learned policy π.
for $t = t_1, \ldots, t_2$ **do**
  Measure $E(t)$ and $\bar{\xi}^t$ to define s^t
  Apply $P_b = \pi(t, s)$ to the battery
end for

Every days the set $\mathcal{D}$ is enriched and therefore Algorithm 8 retrained before the newly learned policy is applied to the next day with Algorithm 9.

5.5 Numerical example

The buying and selling prices of electricity are known: the selling price is set to 0, and the buying price is the real spot price collected over the 2-year period 2022-01-22 to 2024-04-22. Only the consumption and production are stochastic, and all the measures are historical data from a mainly commercial building in Solaize (France).

5.5.1 MPC's updating parameter setting

The MPC control strategy only requires setting the updating horizon h, i.e., the time interval at which the EMS re-computes a prediction and optimization over a 24-hour horizon. For this numerical example, we have set $h = 30$ minutes.

5.5.2 Scenario-based methods tuning

5.5.2.1 Scenario generation

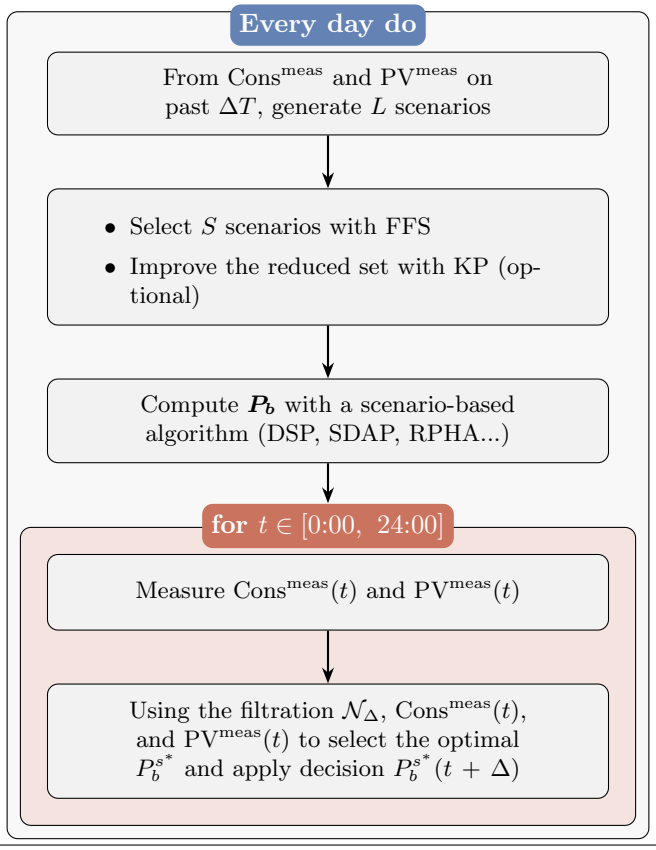
To perform stochastic optimization, it is essential to generate a sufficient number of scenarios to capture daily variability. Leveraging historical data from the system we study, we apply the method proposed by [80] inspired by [4], to produce plausible scenarios that reflect the underlying distribution of the measurements.
In the following methods, we aim to strike a balance between accurately representing uncertainties and maintaining computational tractability. More specifically, this involves controlling the number of scenarios to avoid an exponential increase in complexity. A common approach consists in generating a large set of L equiprobable scenarios and then selecting a reduced subset of $S < L$ scenarios such that the Wasserstein distance to the original distribution is minimized. For this selection step, we use the Fast Forward Selection (FFS) algorithm (see Algorithm 2.1 in [64]). To further improve the quality of the reduced scenario tree, we refine this set by minimizing the nested distance [97] between the reduced and original trees. This refinement is achieved using a boosted version of Kovacevic and Pichler's reduction tree method (KP), as described in Algorithm 3 of [86], and originally introduced in [70].
In the following, we generate an initial set of 10 000 scenarios and reduce it to an optimized set of 225 scenarios for RPHA. The DSP model is simpler to solve and can directly handle 40 000 scenarios. Due to the limitations of our computational setup and the quadratic programming solver, the SDAP algorithm for solving the problem (DRO) struggles to handle a large number of scenarios, mainly because of the projection step onto the ambiguity set (see Definition 12). Therefore, we restricted our experiments to only $S = 100$ and $L = 400$ scenarios.

5.5.2.2 Rolling-horizon framework

The rolling-horizon we use is 24 hours, with $t_1 =$ 0:00 and $t_2 =$ 24:00, corresponding to a typical daily cycle of production and consumption, and thus of battery charging and discharging. Empirically, shorter horizons have been found to yield inferior results. The interval between two steps is $\Delta = 10$ minutes.

The control algorithm is described in the Process 1 above, where $\text{Cons}^{\text{meas}}, \text{PV}^{\text{meas}}$ denote, respectively, the electric consumption and photovoltaic production measured at the meter. These variables are deterministic in the sense that they represent a particular realization of a stochastic process.

Process 1 Decision process for scenario-based methods



5.5.2.3 Tuning hyperparameters

The optimization involves the tuning of hyperparameters: α for VSP and θ for DRO. They are selected through a cross-validation procedure in which Process 1 is run over a 60-day period, from 2024-05-04 to 2024-07-03, for various values of θ and α and with and without the optional step KP. The ground truth for electrical consumption and production is provided by the measured data $\text{Cons}^{\text{meas}}$ and PV^{meas}, respectively.

VSP Figure 5.2 illustrates the impact of the combination of FFS and KP and identifies the best value of the hyperparameter α.

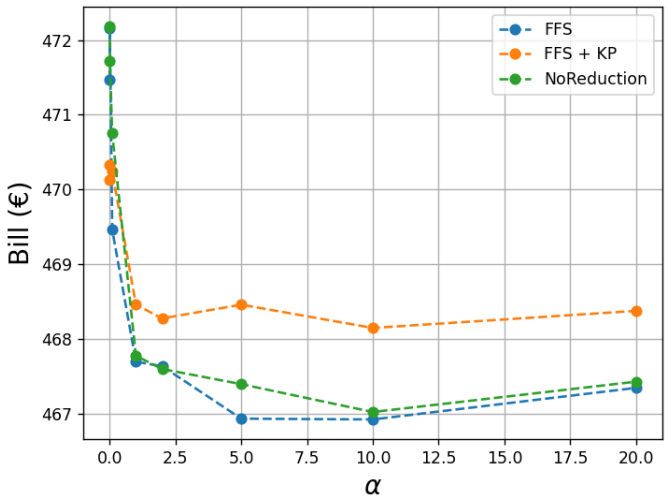


Figure 5.2: Cross-validation VSP: Bill values over a 60 day period for different values of α, with or without the scenario reduction algorithm KP.

For VSP, it is preferable to apply only the FFS reduction method. Attempting to further align the reduced set with the original one using the KP algorithm does not improve performance and may even degrade it by constraining the diversity of the scenarios. It is most likely because our scenario trees are too deep and branch too quickly. However, for problems with fewer stages, this would likely not be the case, and KP would probably constitute a very good option.

DRO The results of this cross-validation are shown in Figure 5.3. As the radius of the ambiguity set varies, the level of robustness is adjusted, and there exists the best value of θ that minimizes the electricity bill. Figure 5.3 also illustrates that the combination of FFS and KP yields better performance compared to using FFS alone, validating thus the interest in using KP.

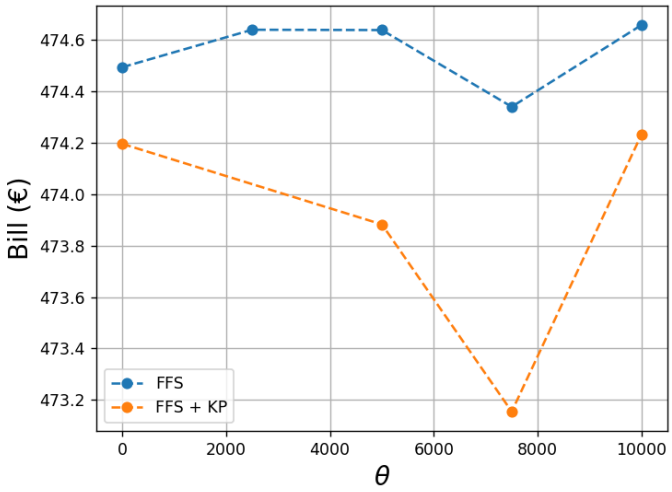


Figure 5.3: Cross-validation DRO: Bill values over a 60 day period for different values of θ and with or without the scenario reduction algorithm KP.

5.5.3 Out-of-sample tests

Finally, we evaluate and compare the performance of the various models—VSP (RPHA), SP, DSP, MPC, RL, and DRO (SDAP)—on the problem over a two-year period, from 2022-01-22 to 2024-01-22.

Figure 5.4 presents the evolution of the performance ratio η, defined as

$$\eta(\text{Day}) := 100 \times \frac{\rho - \text{Bill}}{\rho} \tag{5.21}$$

where ρ represents the battery-less electrical bill:

$$\rho := \int_{t_0}^{t_f} \mathrm{p}_r^c(t) \max\left(\mathrm{Cons}^{\mathrm{meas}}(t) - \mathrm{PV}^{\mathrm{meas}}(t), 0\right) + \mathrm{p}_r^d(t) \min\left(\mathrm{Cons}^{\mathrm{meas}}(t) - \mathrm{PV}^{\mathrm{meas}}(t), 0\right) dt. \tag{5.22}$$

Figure 5.4 highlights the strong performance of DSP, which outperforms all other models. The ideal performance that would be achieved with perfect foresight (solving a fully deterministic problem [18, Section 4] where both production and consumption are perfectly known) is a bill reduction of 12.5%. Thus, DSP reaches 62.5% of the perfect score. Its strong performance is likely due to its ability to leverage a very large number of scenarios, resulting in a highly robust solution. Besides it minimizes the expected cost over all the determinist solutions, making this model highly robust. Both the VSP and RL models demonstrate superior results compared to the classical SP and MPC approaches. Notably, the RL model achieves competitive performance without requiring extensive fine-tuning or cross-validation, in contrast to other methods. At the begining RL does not train on a large dataset $\mathcal{D}$ this is why it displays poor results until a bit less than 200 days are trained on. The SP model outperforms the DRO

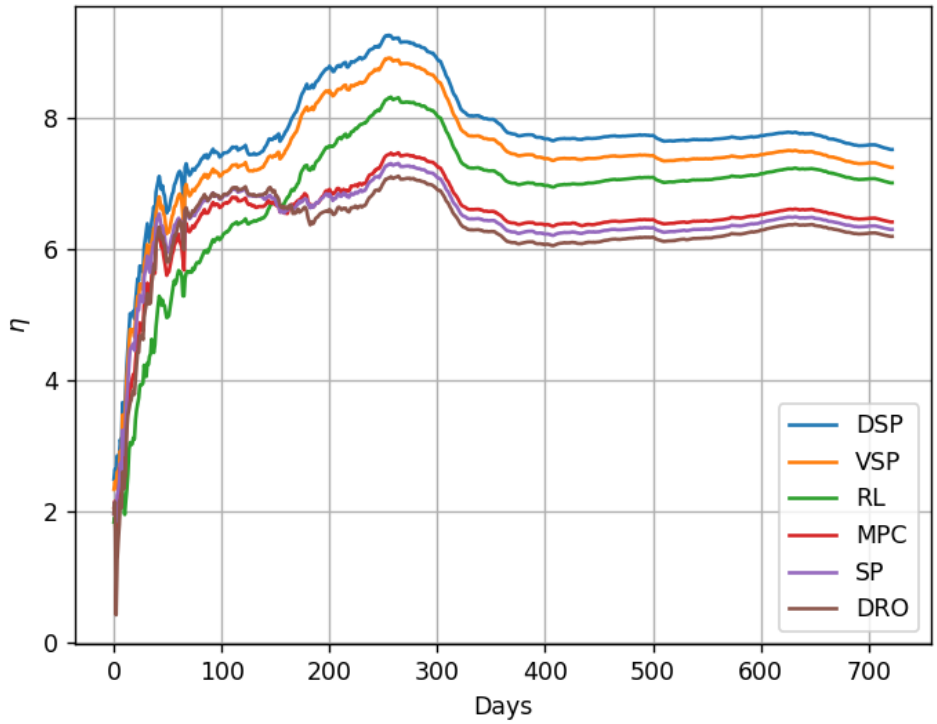


Figure 5.4: Percentage reduction in electricity bill relative to storage-less baseline over two years for models: DSP, VSP (RPHA), SP, EMS, and DRO (SDAP).

approach in terms of bill reduction, this can be attributed to its implementation via PHA with a larger number of scenarios, leading to a more robust solution. In comparison, the SDAP-based DRO model is limited in that it only adjusts the scenario probabilities, leaving the scenario values unchanged. As a result, the model's ability to fine-tune robustness is constrained by the initial scenario set, potentially limiting its overall effectiveness.

An interesting observation is the contrasting efficiency trends during the summer of 2022: the performance of VSP and RL improve, while the standard SP's efficiency declines. This period coincided with unusually high spot electricity prices in France, driven by limited availability of nuclear power plants and elevated gas prices following the Russian invasion of Ukraine. Under such conditions, an effective control strategy must adopt a risk-averse stance to minimize costly electricity consumption. In this regard, the proposed robust approaches demonstrate greater risk aversion than the standard SP and yields better performance compared to the MPC strategy.

We define four criteria to help explain the differences in model performance, where $P_d := -\min(\rho_d P_b^{\text{meas}}, 0)$ and $P_c := \max(P_b^{\text{meas}}/\rho_c, 0)$:

- **Autoproduction gain ratio:** the amount of energy discharged from the battery that helps meet the demand.

This gain is defined as

$$\texttt{PG} = 100 \times \frac{\int \texttt{min}\{P_d(t), \text{Cons}^{\text{meas}}(t) - \text{PV}^{\text{meas}}(t)\}\mathbb{1}_{\mathcal{C}_1}(t)\text{d}t}{\int \text{Cons}^{\text{meas}}(t)dt}. \tag{5.23}$$

Where, $\mathcal{C}_1 : \text{PV}^{\text{meas}}(t) < \text{Cons}^{\text{meas}}(t)$.

- **Autoconsumption gain ratio:** the amount of surplus photovoltaic energy that is effectively stored in the battery. This gain is defined as

$$\texttt{CG} = 100 \times \frac{\int \texttt{max}\{P_c(t), \text{PV}^{\text{meas}}(t) - \text{Cons}\}\mathbb{1}_{\mathcal{C}_2}(t)dt}{\int \text{PV}^{\text{meas}}(t)dt}. \tag{5.24}$$

Where, $\mathcal{C}_2 : \text{PV}^{\text{meas}}(t) > \text{Cons}^{\text{meas}}(t)$.

- **Discharging error ratio:** the amount of energy discharged from the battery that exceeds the energy need, and is somehow spoiled. This ratio is defined as

$$\texttt{DE} = 100 \times \frac{\int \left(P_d(t) - (\text{Cons}^{\text{meas}}(t) - \text{PV}^{\text{meas}}(t))\right)\mathbb{1}_{\mathcal{C}_3}(t)dt}{\int \text{Cons}^{\text{meas}}(t)dt}. \tag{5.25}$$

Where, $\mathcal{C}_3 : \text{PV}^{\text{meas}}(t) < \text{Cons}^{\text{meas}}(t)$ and $\text{Cons}^{\text{meas}}(t) - \text{PV}^{\text{meas}}(t) < P_d(t)$.

- **Grid charging ratio:** the amount of energy charged to the battery directly from the grid. This ratio is defined as

$$\texttt{GC} = 100 \times \frac{\int (P_c(t) - (\text{PV}^{\text{meas}}(t) - \text{Cons}^{\text{meas}}(t)))\mathbb{1}_{\mathcal{C}_4}(t)dt}{\int \text{PV}^{\text{meas}}(t)dt}. \tag{5.26}$$

Where, $\mathcal{C}_4 : \text{PV}^{\text{meas}}(t) > \text{Cons}^{\text{meas}}(t)$ and $\text{PV}^{\text{meas}}(t) - \text{Cons}^{\text{meas}}(t) < P_c(t)$.

These criteria are integrated over the two year period and these values are reported in Table 5.1.

Table 5.1: Evaluation of models according to four performance criteria. Values in parentheses indicate the percentage difference relative to MPC.

Method	CG	PG	DE	GC
DSP	1.2900 (49.7%)	1.4437 (12.2%)	0.0378 (-65.3%)	2.1109 (-9.8%)
VSP	0.9248 (7.3%)	1.3086 (1.7%)	0.0783 (-28.1%)	2.2602 (-3.4%)
RL	0.9189 (6.6%)	1.1824 (-8.1%)	0.0649 (-40.5%)	2.0080 (-14.2%)
MPC	0.8620 (0.0%)	1.2867 (0.0%)	0.1090 (0.0%)	2.3400 (0.0%)
DRO	0.8850 (2.7%)	1.2715 (-1.2%)	0.1135 (4.1%)	2.2927 (-2.0%)
SP	0.8811 (2.1%)	1.2883 (0.1%)	0.1223 (12.2%)	2.3593 (+0.8%)

DSP achieves the highest autoproduction and autoconsumption gain ratios among all methods. This indicates that it stores surplus photovoltaic energy more effectively and makes better use of the battery to cover consumption when needed. Moreover, DSP has the lowest autoproduction loss ratio, meaning it makes fewer errors by discharging

the battery when it is not necessary, compared to the other models. In addition, except for RL, DSP also has the smallest autoconsumption loss ratio, which suggests that it rarely charges the battery when there is insufficient solar production—therefore, it tends to purchase less electricity from the grid. According to Table 5.1, the performance of scenario-based models is primarily driven by their ability to capture gains, whereas the performance of the RL model stems from its ability to limit losses. Indeed, RL exhibits low gain ratios, but also notably low loss ratios.

Chapter 6

Concluding remarks and future work

This thesis addresses multistage decision-making through the lens of multistage stochastic optimization, with a particular focus on the **robustness** of various modeling approaches. These problems are inherently challenging due to the exponential growth in dimensionality caused by the interplay between decisions and uncertainties over time. Reducing this complexity while preserving the essential characteristics of the solution space and optimal value is crucial for real-world applications. To this end, this work introduces a novel methodology based on Wasserstein barycenters for **scenario tree reduction**, aiming to make multistage models more tractable and practically usable. Beyond the theoretical advancements outlined in this thesis recalled below, the proposed methodologies have been successfully applied to industrial energy management problems at IFPEN. Each major research direction pursued during this PhD has led to original scientific contributions, resulting in two published papers and two additional manuscripts currently under review in peer-reviewed journals. To promote transparency and reproducibility, the associated algorithms have been released as open-source Python packages (`https://dan-mim.github.io/`). As a further testament to the practical impact of this work, it has enabled the development of a fully optimized pipeline for industrial energy management systems, which is already integrated within IFPEN's softwares. This integration of cutting-edge research with real-world applications highlights the thesis's dual focus on methodological innovation and industrial relevance, a shared commitment of IFPEN and Mines Paris PSL.

In the pursuit of efficient solutions to multistage stochastic optimization problems, this thesis focuses on enhancing the scenario tree reduction algorithm developed by Kovacevic and Pichler [70]. While this algorithm is recognized for producing high-quality reduced scenario trees that preserve an adherence between the optimal values and solutions of both original and reduced multistage stochastic programs, its computational cost remains prohibitive for large-scale instances commonly encountered in industrial settings. To address this limitation, we conducted an in-depth analysis of the algorithm's structure, revisiting the foundational work [70]. We identified its computational bottleneck: the repeated solution of large-scale linear programs (LPs) at each iteration. Crucially, we discovered that these structured LPs correspond to a well-studied class of problems in machine learning and data science; namely, **Wasserstein barycenter problems**. However, after examination, we have noticed that existing methods for solving large-scale Wasserstein barycenters are predominantly inexact, which limits their applicability in contexts requiring high precision. Moreover, their approximations rely on hyperparameters that must be tuned for each WB computation, whereas in our setting we need to compute a very large number of WBs rapidly. This

observation revealed a broader need for exact and scalable algorithms for barycenter computation, extending beyond our initial goal of scenario tree reduction. To meet this need, we developed a new method for computing Wasserstein barycenters in a broad setting—including balanced and unbalanced formulations. Our approach, the Method of Averaged Marginals (MAM), is built upon the Douglas-Rachford operator splitting algorithm. MAM computes barycenter of measures via the marginals of transport plans and is designed to be simple to implement, with support for parallelized deterministic or randomized computations. This robust and versatile method not only improves the scenario tree reduction of Kovacevic and Pichler but also advances the state-of-the-art in computing Wasserstein barycenters and opens new possibilities in domains where exact barycenters are critical, such as medical image analysis and other applications in machine learning. As detailed in Chapter 2, MAM is capable of solving a novel class of unbalanced optimal transport problems, introduced also in our paper published in the SIAM Journal on Mathematics and Data Sciences (SIMODS) [85]. Further extending this line of research, our second paper—accepted for publication in the Pacific Journal of Optimization (PJO) [84], in a special issue honoring Professor R. Tyrrell Rockafellar—presents an innovative manner for computing constrained Wasserstein barycenters, where constraints are imposed directly on the barycenter rather than on the transport plans. Our approaches for computing constrained Wasserstein barycenters can handle convex and nonconvex constraints as long as projecting onto the set (defined by this constraints) is a simple task. These contributions, discussed in Chapter 3, opens promising avenues for scenario reduction, image processing, and other applications requiring structured Wasserstein barycenter solutions.

After completing our research on Wasserstein barycenters, we returned to our initial motivating application: scenario tree reduction for multistage stochastic programs. As detailed in Chapter 4, we integrated our barycenter computation tools into the algorithm of Kovacevic and Pichler, and benchmarked the enhanced method across various multistage scenario tree configurations. Our experiments demonstrate a significant speedup—exceeding 9000%—in computation time compared to the original algorithm, while maintaining the same level of accuracy in minimizing the nested distance. This improvement highlights the practical value of exploiting the structure of the LPs subproblems in the scenario reduction workflows. A manuscript presenting this innovative approach to scenario tree reduction is currently under review at the Annals of Operations Research [86].

With an efficient algorithm for reducing multistage scenario trees in hand, we turned our attention to the core application of this thesis: the optimization of a real-world energy management system (EMS) developed at IFPEN. As presented in Chapter 5, we explored several modeling approaches for this multiperiod decision problem under uncertainty. These include Model Predictive Control (MPC), risk-neutral and risk-averse multistage stochastic models, distributionally robust optimization, as well as reinforcement learning with scenario-based formulations. To solve these models, we deployed a range of advanced optimization techniques, including the Progressive Hedging Algorithm [104], Regularized Progressive Hedging [80], Scenario Decomposition with Alternating Projections [44], among others. This comprehensive modeling and algorithmic exploration has led to a scientific manuscript currently under revision in a reputable engineering journal [87], highlighting both the methodological depth and the industrial relevance of our work .

6.1 Participation in scientific events and valorization of our research

All the slides and codes can be found on https://dan-mim.github.io/publications/.

The MAM algorithm has been presented in several scientific events:

- **ISMP 2024** – Int. Symposium on **Mathematical Programming** *Computing Wasserstein Barycenters via Operator Splitting* (https://ismp2024.gerad.ca/);
- **EUROPT 2024** – *New Approach to Optimal Transport problems* (with W. de Oliveira) (https://europt2024.event.lu.se/);
- **PGMO 2023** – the annual conference of the Optimization, OR, and Data Science program of the FMJH (Fondation Mathématiques Jacques Hadamard) (https://smf.emath.fr/evenements-smf/pgmo-days-2023);
- **CIROQUO** – Presentation of a poster at the Consortium in Applied Mathematics (https://ciroquo.ec-lyon.fr/evenements.html);
- Presentation of a poster during the DATA IA days of CentraleSupelec. (https://www.dataia.eu/).

The extensions on the constraint WB have been presented in:

- **ISMP 2024** – Int. Symposium on **Mathematical Programming** *Computing Optimal Transport problems* (https://ismp2024.gerad.ca/);
- **ICSP 2025** – Int. Conference on Stochastic Programming *How can constraint Optimal Transport reshape ML?* (with G. Sempere) (https://icsp2025.org/).

The work on the scenario tree reduction has been presented in :

- **PGMO 2024** – Gaspard Monge Program Days (EDF/INRIA) *Boosting Scenario Tree Reduction* (https://www.fondation-hadamard.fr/fr/articles/2024/02/21/pgmodays-2024/);
- **ICSP 2025** – Int. Conference on Stochastic Programming *How Optimal Transport can sharpen multi-stage decisions: Boosting scenario tree algorithms* (https://icsp2025.org/).

The comparison of robustness between reinforcement learning and other stochastic programming model has been presented in:

- **ICCOPT 2025** – Int. Conf. on Continuous Optimization *Optimization framework for Energy Management Systems: RL vs Stochastic Programming* (https://sites.google.com/view/iccopt2025/home).

Notable conferences and events I attended within these years include:

- **NeurIPS Paris** (https://scai.sorbonne-universite.fr/public/events/view/4ad2190f2c212abfd60f/8)

Key courses I participated in:

- **MVA/MASH** Course on Computational Optimal Transport by Gabriel Peyré (https://www.master-mva.com/cours/computational-optimal-transport/)
- **PSL** week on Stochastic Optimization by Welington de Oliveira (https://www.oliveira.mat.br/teaching)

Courses I instructed in 2023, 2024 and 2025:

- **Mines Paris**, *Mathematics for Data Science* (L3), Centre de Morphologie Mathématique (https://www.cmm.minesparis.psl.eu/), with Bruno Figliuzzi and Chloé-Agathe Azencott.

6.2 Future research directions

Several topics could be considered in the continuation of this work:

- An interesting improvement to MAM would be to replace the Euclidean distance with Bregman distance functions in the Douglas-Rachford scheme and leverage the results of [30] since the authors give explicit steps. The research process is relatively straightforward: adapt the current method with minor modifications to incorporate [30], and if the preliminary results are promising, pursue a deeper mathematical analysis.
- Another avenue would be to enhance MAM using a Douglas–Rachford splitting scheme with adaptive step sizes, potentially by monitoring the distance to the set $\mathcal{B}$.
- Implementing MAM on GPUs could significantly improve performance. While the current implementation on GitHub [81] is CPU-based, a GPU-enabled version is under consideration.
- A promising line of inquiry is to explore a new approach for accelerating MAM Algorithm 1 in the unbalanced barycenter setting, inspired by the Frank–Wolfe algorithm (FW) [9]. This method has been investigated in the context of Sinkhorn's algorithm [78]; however, applying it to the unbalanced formulation proposed in eq. (2.11) could yield novel insights. The idea is to linearize the squared Euclidean distance in eq. (2.11), making minimization via FW explicit. This could potentially bypass the most computationally expensive step in MAM, namely the projection onto the simplex. The key question would then be to study the convergence trade-off: while each FW iteration is fast, the total number of iterations required for convergence can be substantial.
- Studying barycenters with the Hellinger distance [121] could also be worthwhile, as they may behave similarly to the Unbalanced Wasserstein barycenters (UWB) [76] proposed in Chapter 2. While UWB may be more general and suitable for imaging problems, the Hellinger formulation could be simpler to compute and does not require tuning a hyperparameter—an advantage in some settings, though it can also be a limitation.
- Similar to our extension of the Wasserstein barycenter problem, the *best barycenter problem* [48] presents another extension with strong industrial relevance (e.g., at IFPEN in the context of reduced basis methods

for PDEs). This problem has been studied for large-scale PDE settings to learn the space, but due to its complexity, it has only been industrially implemented for 1D barycenters. More recently, a theory for higher dimensions [7] has been proposed but it leverages a regularization that may significantly affect solution accuracy; it also uses automatic differentiation that can accumulate errors here. It is worth investigating whether MAM's Douglas–Rachford steps could be adapted to this specialized challenge.

- Further improvements in combining Wasserstein barycenters with scenario tree reduction are conceivable. For instance, one could experiment with warm-starting MAM through multiple node-dependent barycenter problems.

- It would probably be possible to propose a scenario reduction method that unlike the approach in Chapter 4, does not require an initial filtration and instead operates directly on the original tree. The idea would be to apply the Kovacevic–Pichler algorithm to the original tree, with the reduced tree preserving the same filtration, but replacing the classical Wasserstein barycenter in the recursion with a constrained Wasserstein barycenter (see Chapter 3) to trim the filtration of the subtrees. The constraint would require finding a barycenter of the original subtrees with a reduced number of edges (supports) compared to the original tree.

- A new type of barycenter, which one might call a *Process Barycenter*, could be defined using the Nested/Process distance of G. Pflug and A. Pichler [97]. While formulating such a concept is straightforward, further work would be required to determine whether it offers novel insights, mathematical properties, or computational advantages for scenario tree reduction.

- Another potential application of scenario tree reduction lies in recent reinforcement learning techniques, such as policy gradient with tree expansion [41]. In tree-based exploration, the search typically relies on heuristics to select promising nodes to build the trees; here, tree reduction could be applied directly. The new tree reduction approach we proposed in Chapter 4 is particularly fast for very large but shallow trees, which fits this context well.

- We have provided a rigorous review comparing different models for multistage stochastic problems in Chapter 5. Although we did not employ models that violate the assumptions of our problem class (e.g., no stagewise independence), it could be valuable to use such models as heuristics. One example is dynamic programming with Stochastic Dual Dynamic Programming (SDDP) [91], which remains widely used in practice.

- In Chapter 5, we noted that SDAP is computationally expensive, primarily due to the projection onto the Wasserstein ambiguity set. This projection problem arises in many contexts, so developing an efficient method for it would be of broad interest. The nature of the problem suggests that it might be reformulated within a Douglas–Rachford framework.

- Replacing the computationally demanding Wasserstein distance in DRO with the Maximum Mean Discrepancy (MMD) [61] could be another interesting direction. MMD is easier to compute, which could allow DRO to handle larger scenario trees. The challenge would shift to kernel selection, which, while nontrivial, might still be easier to address.

- Another limitation of the WDRO model we adopt in Chapter 5 is the fixed support: our study optimizes only the probabilities, but allowing the support to be optimized would increase generality and flexibility. At present, however, this remains a large-scale NP-hard problem.

- Scenario tree reduction has proven to be an effective tool for improving the adaptability of classical stochastic optimization methods. It would be interesting to analyze tree structures encountered in multistage DRO. In

our study, the trees have many stages and tend to adopt a fan-like structure early on. Using large binary trees, for example, might help preserve generality in DRO, though this would likely lead to combinatorial explosion and should first be tested with fewer stages. Sampling at hourly intervals and performing projections could help mitigate this issue.

- The introduction of RPHA by Malisani et al. [80] motivates a generalization of the work by Blanchet et al. [19], which shows an equivalence between DRO and regularized optimization for classification. Investigating the theoretical proximity between RPHA and WDRO in the multistage setting would be of interest.
- In our reinforcement learning experiments of Chapter 5, we did not employ a state-of-the-art method; instead, we used classical Q-learning techniques. Despite this, the model performed well, suggesting that applying more advanced RL algorithms could yield even better results and provide a stronger challenge to DSP. For example, with a larger dataset we could implement a neural RL model.

Bibliography

[1] M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, Jan 2011.

[2] J. M. Altschuler and E. Boix-Adsera. Wasserstein barycenters can be computed in polynomial time in fixed dimension. *The Journal of Machine Learning Research*, 22(1):2000–2018, 2021.

[3] J. M. Altschuler and E. Boix-Adserà. Wasserstein barycenters are NP-Hard to compute. *SIAM Journal on Mathematics of Data Science*, 4(1):179–203, 2022.

[4] L. Amabile, D. Bresch-Pietri, G. El Hajje, S. Labbé, and N. Petit. Optimizing the self-consumption of residential photovoltaic energy and quantification of the impact of production forecast uncertainties. *Advances in Applied Energy*, 2:100020, 2021.

[5] B. Analui and G. C. Pflug. On distributionally robust multiperiod stochastic optimization. *Computational Management Science*, 11:197–220, 2014.

[6] E. Anderes, S. Borgwardt, and J. Miller. Discrete Wasserstein barycenters: optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84(2):389–409, 2016.

[7] B. Battisti, T. Blickhan, G. Enchery, V. Ehrlacher, D. Lombardi, and O. Mula. Wasserstein model reduction approach for parametrized flow problems in porous media. *ESAIM: Proceedings and Surveys*, 73:28–47, 2023.

[8] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2nd edition, 2017.

[9] A. Beck. *First-order methods in optimization*. SIAM, 2017.

[10] Mathias Beiglböck, Pierre Henry-Labordere, and Friedrich Penkner. Model-independent bounds for option prices: A mass transport approach. *Finance and Stochastics*, 17:477–501, 2013.

[11] R. Bellman. Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228–239, 1958.

[12] F. Beltran, W. de Oliveira, and E. C. Finardi. Application of scenario tree reduction via quadratic process to medium-term hydrothermal scheduling problem. *IEEE Transactions on Power Systems*, 32(6):4351–4361, 2017.

[13] F. Beltrán, E. C. Finardi, and W. de Oliveira. Two-stage and multi-stage decompositions for the medium-term hydrothermal scheduling problem: A computational comparison of solution techniques. *International Journal of Electrical Power & Energy Systems*, 127:106659, 2021.

[14] F. Beltrán, E. C. Finardi, G. M. Fredo, and W. de Oliveira. Improving the performance of the stochastic dual dynamic programming algorithm using chebyshev centers. *Optimization and Engineering*, pages 1–22, 2020.

[15] J. D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[16] D. Bertsekas. Dynamic programming and optimal control: Volume i, 2012.

[17] D. P. Bertsekas. *Convex Optimization Algorithms*. Number 1st. Athena Scientific, 2015.

[18] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Science & Business Media, 2011.

[19] J. Blanchet, Y. Kang, and K. Murthy. Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

[20] J. F. Bonnans. *Convex and Stochastic Optimization*. Universitext. Springer International Publishing, Cham, 2019.

[21] S. Borgwardt. An LP-based, strongly-polynomial 2-approximation algorithm for sparse Wasserstein barycenters. *Operational Research*, 22(2):1511–1551, Apr 2022.

[22] S. Borgwardt and S. Patterson. Improved linear programs for discrete barycenters. *INFORMS Journal on Optimization*, 2(1):14–33, 2020.

[23] S. Borgwardt and S. Patterson. On the computational complexity of finding a sparse Wasserstein barycenter. *Journal of Combinatorial Optimization*, 41(3):736–761, 2021.

[24] D. Bousnina and G. Guerassimoff. Deep reinforcement learning for optimal energy management of multi-energy smart grids. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 15–30. Springer, 2021.

[25] A. I. Brodt. Min-mad life: A multi-period optimization model for life insurance company investment decisions. *Insurance: Mathematics and Economics*, 2(2):91–102, 1983.

[26] J.B. Caillau, R. Ferretti, E. Trélat, and H. Zidani. Chapter 15 - an algorithmic guide for finite-dimensional optimal control problems. In *Numerical Control: Part B*, volume 24 of *Handbook of Numerical Analysis*, pages 559–626. Elsevier, 2023.

[27] G. Carlier, A. Oberman, and E. Oudet. Numerical methods for matching for teams and Wasserstein barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1621–1642, Nov 2015.

[28] P. Carpentier, J.-P. Chancelier, G. Cohen, and M. De Lara. *Stochastic Multi-Stage Optimization: At the Crossroads between Discrete Time Stochastic Control and Stochastic Programming*, volume 75 of *Probability Theory and Stochastic Modelling*. Springer International Publishing, Cham, 2015.

[29] P. Carpentier, J.-P. Chancelier, G. Cohen, M. De Lara, and P. Girardeau. Dynamic consistency for stochastic optimal control problems. *Annals of Operations Research*, 200:247–263, 2012.

[30] A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287, 2016.

[31] Z. Chen and Z. Yan. Scenario tree reduction methods through clustering nodes. *Computers & Chemical Engineering*, 109:96–111, 2018.

[32] D. Cole, H. Sharma, and W. Wang. Contextual reinforcement learning for offshore wind farm bidding. *arXiv preprint arXiv:2312.10884*, 2023.

[33] P. L. Combettes and J. Eckstein. Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions. *Mathematical Programming*, 168(1):645–672, 2018.

[34] P. L. Combettes and J.-C. Pesquet. Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping. *SIAM Journal on Optimization*, 25(2):1221–1248, 2015.

[35] L. Condat. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, 158(1):575–585, Jul 2016.

[36] Y. Cui and J.-S. Pang. *Modern Nonconvex Nondifferentiable Optimization*. SIAM, 2022.

[37] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[38] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Beijing, China, 22–24 Jun 2014. PMLR.

[39] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. *International Conference on Machine Learning*, 32(2):685–693, 2014.

[40] M. Cuturi and G. Peyré. A smoothed dual approach for variational Wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016.

[41] G. Dalal, A. Hallak, G. Thoppe, S. Mannor, and G. Chechik. Policy gradient with tree expansion. 2024.

[42] G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955.

[43] W. de Oliveira. The ABC of DC programming. *Set-Valued Var. Anal.*, 28(4):679–706, 2020.

[44] W. de Oliveira. Risk-averse stochastic programming and distributionally robust optimization via operator splitting. *Set-Valued and Variational Analysis*, 29(4):861–891, 2021.

[45] W. de Oliveira and J. C. de Oliveira Souza. A progressive decoupling algorithm for minimizing the difference of convex and weakly convex functions. *Journal of Optimization Theory and Applications*, 204(3):36, Jan 2025.

[46] W. de Oliveira, C. Sagastizábal, D. D. J. Penna, M. E. P. Maceira, and J. M. Damázio. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. *Optimization Methods & Software*, 25(6):917–936, 2010.

[47] W. de Oliveira, C. Sagastizábal, D. D. J. Penna, M. E. P. Maceira, and J. M. Damázio. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. *Optimization Methods and Software*, 25(6):917–936, 2010.

[48] M.-H. Do, J. Feydy, and O. Mula. Approximation and structured prediction with sparse Wasserstein barycenters. *arXiv preprint arXiv:2302.05356*, 2023.

[49] Yan Dolinsky and H. Mete Soner. Robust hedging with proportional transaction costs. *Finance and Stochastics*, 18:327–347, 2014.

[50] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.

[51] C. Duan, W. Fang, L. Jiang, L. Yao, and J. Liu. Distributionally robust chance-constrained approximate ac-opf with Wasserstein metric. *IEEE Transactions on Power Systems*, 33(5):4924–4936, 2018.

[52] J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenario reduction in stochastic programming. *Mathematical Programming*, 95:493–511, 2003.

[53] J. Eckstein and D. P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1–3):293–318, Apr 1992.

[54] N. C. P. Edirisinghe. Multiperiod portfolio optimization with terminal liability: Bounds for the convex case. *Computational Optimization and Applications*, 32:29–59, 2005.

[55] P. M. Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[56] A. Fu, J. Zhang, and S. Boyd. Anderson accelerated Douglas–Rachford splitting. *SIAM Journal on Scientific Computing*, 42(6):A3560–A3583, Jan 2020.

[57] A. Galichon, P. Henry-Labordere, and N. Touzi. A stochastic control approach to no-arbitrage bounds given marginals, with an application to lookback options. 2014.

[58] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[59] H. Golpira and S. A. R. Khan. A multi-objective risk-based robust optimization approach to energy management in smart residential buildings under combined demand and supply uncertainty. *Energy*, 170:1113–1129, 2019.

[60] A. Gramfort, G. Peyré, and M. Cuturi. Fast optimal transport averaging of neuroimaging data. In S. Ourselin, D. C. Alexander, C.-F. Westin, and M. J. Cardoso, editors, *Information Processing in Medical Imaging*, pages 261–272, Cham, 2015. Springer International Publishing.

[61] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

[62] B. He and X. Yuan. On the convergence rate of Douglas-Rachford operator splitting method. *Math. Program.*, 153:715–722, 2015.

[63] F. Heinemann, M. Klatt, and A. Munk. Kantorovich–Rubinstein distance and barycenter for finitely supported measures: Foundations and algorithms. *Applied Mathematics & Optimization*, 87(1):4, Nov 2022.

[64] H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24:187–206, 2003.

[65] H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118:371–406, 2009.

[66] H. Heitsch, W. Römisch, and C. Strugarek. Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17(2):511–525, 2006.

[67] M. Horejšová, S. Vitali, M. Kopa, and V. Moriggia. Evaluation of scenario reduction algorithms with nested distance. *Computational Management Science*, 17(2):241–275, 2020.

[68] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *52nd IEEE Conference on Decision and Control*. IEEE, Dec 2013.

[69] S. Kammammettu and Z. Li. Scenario reduction and scenario tree generation for stochastic programming using Sinkhorn distance. *Computers & Chemical Engineering*, 170:108122, 2023.

[70] R. M. Kovacevic and A. Pichler. Tree approximation for discrete time stochastic processes: a process distance approach. *Annals of Operations Research*, 235(1):395–421, 2015.

[71] M. Y. Lamoudi. *Distributed model predictive control for energy management in buildings*. PhD thesis, Université de Grenoble, 2012.

[72] J. Li, Z. Xu, H. Liu, C. Wang, L. Wang, and C. Gu. A Wasserstein distributionally robust planning model for renewable sources and energy storage systems under multiple uncertainties. *IEEE Transactions on Sustainable Energy*, 14(3):1346–1356, 2022.

[73] Q. Li, X. Meng, F. Gao, G. Zhang, W. Chen, and K. Rajashekara. Reinforcement learning energy management for fuel cell hybrid systems: A review. *IEEE Industrial Electronics Magazine*, 17(4):45–54, 2022.

[74] Z. Li and C. A. Floudas. Optimal scenario reduction framework based on distance of uncertainty distribution and output performance: I. single reduction via mixed integer linear optimization. *Computers & Chemical Engineering*, 70:50–66, 2014.

[75] Z. Li and Z. Li. Linear programming-based scenario reduction using transportation distance. *Computers & Chemical Engineering*, 88:50–58, 2016.

[76] M. Liero, A. Mielke, and G. Savaré. Optimal transport in competition with reaction: The hellinger–kantorovich distance and geodesic curves. *SIAM Journal on Mathematical Analysis*, 48(4):2869–2911, 2016.

[77] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *Society for Industrial and Applied Mathematics*, 19(6):964–979, 1979.

[78] G. Luise, S. Salzo, M. Pontil, and C. Ciliberto. Sinkhorn barycenters with free support via frank-wolfe algorithm. *Advances in Neural Information Processing Systems*, 32, 2019.

[79] P. Malisani. Interior point methods in optimal control. *ESAIM: Control, Optimisation and Calculus of Variations*, 30(59), 2024.

[80] P. Malisani, A. Spagnol, and V. Smis-Michel. Robust stochastic optimization via regularized PHA: Application to energy management systems. *arXiv preprint arXiv:2411.02015*, 2024.

[81] D. Mimouni. Computing-Wasserstein-Barycenters-MAM. https://github.com/dan-mim/Computing-Wasserstein-Barycenters-MAM, 2025. GitHub repository.

[82] D. Mimouni. Constrained-Optimal-Transport. https://github.com/dan-mim/Constrained-Optimal-Transport, 2025. GitHub repository.

[83] D. Mimouni. Nested_tree_reduction. https://github.com/dan-mim/Nested_tree_reduction, 2025. GitHub repository.

[84] D. Mimouni, W. de Oliveira, and G. M. Sempere. On the computation of constrained Wasserstein barycenters. *Pacific Journal of Optimization*, 2025.

[85] D. Mimouni, P. Malisani, J. Zhu, and W. de Oliveira. Computing Wasserstein barycenters via operator splitting: The method of averaged marginals. *SIAM Journal on Mathematics of Data Science*, 6(4):1000–1026, 2024.

[86] D. Mimouni, P. Malisani, J. Zhu, and W. de Oliveira. Scenario tree reduction via Wasserstein barycenters. *arXiv preprint arXiv:2411.14477*, 2024.

[87] D. Mimouni, J. Zhu, W. de Oliveira, and P. Malisani. A comparative study of multi-stage stochastic optimization approaches for an energy management system. *arXiv preprint*, 2025.

[88] F. Oldewurtel. *Stochastic model predictive control for energy efficient building climate control*. PhD thesis, ETH Zurich, 2011.

[89] F. Pacaud, M. De Lara, J.-P. Chancelier, and P. Carpentier. Distributed multistage optimization of large-scale microgrids under stochasticity. *IEEE Transactions on Power Systems*, 37(1):204–211, 2022.

[90] H. Paulo, T. Cardoso-Grilo, S. Relvas, and A. P. Barbosa-Póvoa. Designing integrated biorefineries supply chain: combining stochastic programming models with scenario reduction methods. In *Computer Aided Chemical Engineering*, volume 40, pages 901–906. Elsevier, 2017.

[91] M. V. F. Pereira and L. M. V. G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1):359–375, 1991.

[92] G. Peyré. Bregmanot. https://github.com/gpeyre/2014-SISC-BregmanOT, 2014. GitHub repository.

[93] G. Peyré and M. Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[94] Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2020.

[95] G. Pflug and A. Pichler. *Multistage Stochastic Optimization*. Springer International Publishing, 2014.

[96] G. C. Pflug. Version-independence and nested distributions in multistage stochastic optimization. *SIAM Journal on Optimization*, 20(3):1406–1420, 2010.

[97] G. C. Pflug and A. Pichler. A distance for multistage stochastic optimization models. *SIAM Journal on Optimization*, 22(1):1–23, 2012.

[98] A. Pichler and M. Weinhardt. The nested Sinkhorn divergence to learn the nested distance. *Computational Management Science*, 19(2):269–293, 2022.

[99] G. Puccetti, L. Rüschendorf, and S. Vanduffel. On the computation of Wasserstein barycenters. *Journal of Multivariate Analysis*, 176(104581), Feb 2020.

[100] X. Qian and Q. Tang. Scenario reduction method based on output performance for condition-based maintenance optimization. *Mechanics*, 23(5):743–749, 2017.

[101] R. T. Rockafellar. Progressive decoupling of linkages in optimization and variational inequalities with elicitable convexity or monotonicity. *Set-Valued and Variational Analysis*, 27:863–893, 2019.

[102] R. T. Rockafellar. Generalizations of the proximal method of multipliers in convex optimization. *Computational Optimization and Applications*, 87(1):219–247, Jan 2024.

[103] R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[104] R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[105] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000.

[106] F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs and Modeling*. Birkhäuser Cham, 2015.

[107] T. Sejourne, G. Peyré, and F. Vialard. Unbalanced optimal transport, from theory to numerics. *Handbook of Numerical Analysis*, 24:407–471, Jan 2023.

[108] G. M. Sempere, W. de Oliveira, and J. O. Royset. A proximal-type method for nonsmooth and nonconvex constrained minimization problems. *Journal of Optimization Theory and Applications*, 204(3):54, Feb 2025.

[109] A. Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[110] A. Shapiro. Distributionally robust modeling of optimal control. *Operations Research Letters*, 50(5):561–567, 2022.

[111] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, 2009.

[112] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, 2009.

[113] D. Simon and A. Aberdam. Barycenters of natural images constrained Wasserstein barycenters for image morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7910–7919, 2020.

[114] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. ii. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974.

[115] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[116] J. E. Smith and R. L. Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006.

[117] J. Sun and M. Zhang. The elicited progressive decoupling algorithm: A note on the rate of convergence and a preliminary numerical experiment on the choice of parameters. *Set-Valued and Variational Analysis*, 29(4):997–1018, Dec 2021.

[118] G. Tartavel, G. Peyré, and Y. Gousseau. Wasserstein loss for image synthesis and restoration. *SIAM Journal on Imaging Sciences*, 9(4):1726–1755, 2016.

[119] Tijmen. affnist. https://www.cs.toronto.edu/ tijmen/affNIST/, 2013.

[120] W. S. van Ackooij and W. de Oliveira. *Methods of Nonsmooth Optimization in Stochastic Programming: From Conceptual Algorithms to Real-World Applications*. International Series in Operations Research & Management Science. Springer Cham, 1 edition, 2025.

[121] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[122] C. Villani. *Optimal transport: old and new*, volume 338. Springer Verlag, 2009.

[123] J. von Lindheim. Simple approximative algorithms for free-support Wasserstein barycenters. *Computational Optimization and Applications*, 85(1):213–246, 2023.

[124] H. Wang and A. Banerjee. Bregman alternating direction method of multipliers. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[125] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[126] J. Watson and D. Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, Jul 2010.

[127] L. Weber, A. Bušić, and J. Zhu. Reinforcement learning based demand charge minimization using energy storage. In *Proceedings of the 62nd IEEE Conference on Decision and Control (CDC)*, pages 4351–4357. IEEE, 2023.

[128] D. Xu, Z. Chen, and L. Yang. Scenario tree generation approaches using k-means and LP moment matching methods. *Journal of Computational and Applied Mathematics*, 236(17):4561–4579, 2012.

[129] Z. Xu, M. A. T. Figueiredo, and T. Goldstein. Adaptive ADMM with Spectral Penalty Parameter Selection. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 718–727. PMLR, 20–22 Apr 2017.

[130] J. Ye and J. Li. Scaling up discrete distribution clustering using ADMM. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5267–5271, 2014.

[131] J. Ye, P. Wu, J. Z. Wang, and J. Li. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65:2317–2332, May 2017.